Proxmox para el aula

Arcadio Ortega Reinoso

Este documento está bajo licencia:

Atribución – No Comercial – Compartir Igual (CC BY-NC-SA 4.0)
Puedes compartir, copiar, redistribuir y adaptar esta obra con fines no comerciales,
siempre que cites al autor y mantengas esta misma licencia para cualquier derivado.

Proxmox

Índice

- Acerca del documento
  - Introducción
  - Objetivos
  - Contenido principal del documento
    - Estrategia
- IPA Server
  - Rocky Linux para IPA Server
  - IPA server
- Proxmox
  - Instalación PROXMOX
  - PROXMOX. Ejemplo máquina virtual
  - Errores de update
- Proxmox - IPA
  - Integrar proxmox y IPA
  - Verifica ipa server los usuario
  - Realm --- Sincronizar usuarios con IPA server
- Nodos
  - Cliente sólo browser (firefox modo kiosk)
  - Añadir un segundo proxmox a ipa server
  - Unir un Ubuntu Server a un IPA Server (FreeIPA) como cliente
  - TrueNas IPA
  - Dar permiso sudo a un usuario en FreeIPA
  - NFS server
- Propuesta sin VLAN para una clase
  - Cluster. Propuesta inicial
  - Cluster
  - Ejemplo de NFS en truenas
  - Gestión usuarios
  - Ver los roles
- Nodos del clúster
  - Eliminar Clusteres y/o sus nodos
  - Dar de baja un clúster
  - Cómo añadir un nuevo nodo a un clúster Proxmox con servidor NTP personalizado
- AULA
  - Servidor previo
  - Script ejecutar en MV

- Notas
- Operaciones en proxmox
- Proxmox. Algunos comandos
- Listar máquinas virtuales
- pvecm status
- qm config <Número de máquina virtual>
- Proxmox Información
- LDAP. Información
- Proxmox. IP
- Planificación clases. Pasos a realizar por el profesor
- Algunas operaciones de interés
- Usando TEMPLATES para generar máquinas virtuales
- MV Desde nfs a local
- Template desde la MV creada anteriormente
- Laboratorio clase
- Otros datos de gestión
- Para el futuro
- ¿Qué añadir?

Acerca del documento

Introducción

Este documento describe una configuración de infraestructura de virtualización utilizando Proxmox VE, almacenamiento gestionado por TrueNAS, y un servidor de gestión de identidades FreeIPA (IPA Server).
La arquitectura propuesta permite crear un entorno robusto para servidores, virtualización y alta disponibilidad.

El presente documento proporciona la base técnica para implementar una infraestructura de virtualización centralizada con Proxmox y TrueNAS, y lo más importante, la integración con FreeIPA y el uso de ACLs de Proxmox para controlar de manera eficiente y granular el acceso de los alumnos a las máquinas virtuales necesarias para sus asignaturas o laboratorios, automatizando gran parte del proceso de asignación y revocación de permisos.

Objetivos

- **Instalar ipa server:**

Los fuentes detallan la instalación de IPA Server, incluyendo la preparación del disco, la configuración de red estática, los paquetes necesarios, la ejecución del instalador `ipa-server-install` con las opciones típicas, la configuración del firewall para abrir los puertos necesarios, y la verificación del estado de los servicios de IPA. Se menciona que es una solución basada en FreeIPA y sus características principales como gestión centralizada de identidades, autenticación Kerberos/LDAP, DNS, NTP, políticas de seguridad, auditoría e interfaz web

- **Instalar TrueNas server:**

Los fuentes describen su rol y requisitos. Se mencionan los requisitos de hardware para TrueNAS CORE (RAM, discos, red) y se pasa directamente a la configuración de almacenamiento con ZFS y su compartición una vez que TrueNAS ya está operativo

- **Instalar proxmox:**

Similar a TrueNAS, los fuentes describen su rol y parte de la configuración inicial. Se mencionan los datos que se configuran durante la instalación (país, hora, IP, FQDN, servidor DNS) y se presenta como la plataforma de virtualización

- **Conectar proxmox con TrueNas vía nfs:**

Los fuentes explican cómo configurar el almacenamiento ZFS en TrueNAS (crear pools y datasets), cómo compartir un dataset vía NFS, cómo configurar el montaje NFS en Proxmox (`/etc/fstab`), cómo configurar la compartición NFS en Proxmox VE, y cómo probar la conexión. Se menciona la desactivación de "root squash" y la configuración de Maproot User/Group en TrueNAS para evitar problemas de permisos

- **Conectar proxmox con Ipa server:**

La integración con IPA para autenticación centralizada es un punto clave. Se detallan los pasos para asegurar que Proxmox use el DNS de IPA, añadir el host Proxmox en IPA, obtener el keytab, configurar Kerberos (`krb5.conf`), instalar el cliente IPA (`freeipa-client`) y otras herramientas como SSSD y LDAP/PAM libs, configurar `nslcd.conf` o SSSD, verificar la sincronización de usuarios, y configurar el Realm LDAP en Proxmox GUI o domains.cfg

- **Crear máquinas virtuales en proxmox:**

Se menciona que Proxmox gestiona VMs y contenedores y se muestran comandos básicos para gestionarlas. El proceso de crear una VM base para convertirla en plantilla usando `cloud-init` es descrito en detalle

- **Generar las máquinas virtuales en directorios nfs de TrueNas:**

Al configurar el almacenamiento NFS en Proxmox, se especifica que puede contener images, lo cual es obligatorio para almacenar discos de VMs. La creación de la plantilla base y la clonación se muestran apuntando al almacenamiento NFS configurado (nfs-vm)

- **Integrar máquinas virtuales en Ipa server:**

Se muestra un script (`cliente_ipa.sh`) para añadir la nueva VM como host en IPA Server antes de ejecutar el script de configuración interna. El script (`setup_clone.sh`) que se

ejecuta dentro de la VM clonada configura la VM como cliente IPA, integrándola con SSSD para autenticación

- **Montar homes desde TrueNas en máquinas virtuales:**

La estrategia menciona el uso de NFS en TrueNAS para almacenar directorios home de usuarios. El script `setup_clone.sh` instala `nfs-common` y añade una entrada en `/etc/fstab` para montar el dataset NFS de TrueNAS (`/mnt/truenasdisk/nfsdata`) en `/mnt/home` dentro de la VM, aunque la línea comentada sugiere que el montaje de mkhomedir a través de PAM/SSSD no se activó, y en su lugar se optó por montar el directorio NFS

- **Crear un clúster en proxmox:**

Se describe la configuración de un clúster. Se mencionan las redes recomendadas (gestión, clúster, almacenamiento). Se proporcionan los comandos `pvecm create` y `pvecm add` para crear y añadir nodos. Se explica la importancia del quórum y cómo verificar el estado con `pvecm status`. Se detallan los pasos para configurar la sincronización NTP con el servidor IPA en los nodos del clúster

- **Intercambiar máquinas virtuales entre nodos de proxmox:**

Se menciona la posibilidad de migrar o clonar VMs entre nodos. El comando qm migrate se muestra como opción, aunque un ejemplo muestra un error común. La estrategia principal para el aula implica copiar discos de NFS a almacenamiento local en el nodo de destino, lo cual está cubierto por el script `migrar_vm_desde_nodo.sh`

- **Crear templates desde máquinas:**

Se muestra el comando `qm template` . Se detalla el proceso de crear una VM base con `cloud-init` y luego convertirla en plantilla

- **Crear clones desde templates:**

Se muestra el comando `qm clone` . Se proporciona un script (crear_`clone.sh`) para automatizar la clonación y configuración básica de red y usuario root

- **Mover máquinas virtuales desde nfs a file system local:**

La necesidad de este paso se explica por las limitaciones de red. Se muestra cómo habilitar el contenido images en el almacenamiento local de Proxmox. El script `migrar_vm_desde_nodo.sh` realiza la copia del disco de la VM desde el almacenamiento NFS de origen (`/mnt/pve/STORAGE_NAME/images/...`) al directorio local (`/var/lib/vz/images/VMID_NUEVO/`) y actualiza la configuración de la VM para usar el disco local

- **Dar permisos específicos a usuarios sobre las máquinas virtuales:**

La gestión de permisos y roles en Proxmox integrados con usuarios de IPA es detallada. Se explican los roles PVEVMUser (encender/apagar, ver estado) y PVEVMAdmin (más gestión). Se muestra cómo asignar permisos usando la GUI de Proxmox o la línea de comandos con `pveum aclmod` / `pveum acl modify`. Se muestra cómo revocar permisos con pveum acldel y se presentan scripts para automatizar la concesión y revocación temporal de permisos

- **Dar solución a un aula de informática vía proxmox con Ipa y TrueNas servers:**

La arquitectura propuesta y la estrategia se basan precisamente en la combinación de estos tres elementos para gestionar de forma centralizada identidades y almacenamiento, permitiendo el despliegue rápido y la gestión granular de VMs para alumnos. Se explica cómo

se utiliza cada componente para lograr este fin (IPA para usuarios/grupos, TrueNAS para almacenamiento centralizado de datos/homes, Proxmox para virtualización y control de acceso). El flujo de trabajo de preparación y ejecución de clases (planificación, preparación por el profesor, despliegue local de VMs, asignación automatizada de permisos, finalización) está descrito en detalle.

Contenido principal del documento

Gestión de Almacenamiento

Se exploran diferentes sistemas de archivos y su adecuación para distintos escenarios, como ZFS, CephFS, GlusterFS, BeeGFS, XFS y Btrfs.
Se detalla la configuración de almacenamiento en TrueNAS utilizando ZFS, incluyendo la creación de pools y datasets.
Se explica cómo compartir este almacenamiento a través de protocolos como NFS e iSCSI, y cómo configurar estos almacenamientos en Proxmox VE.
Se diferencia entre el acceso a nivel de archivo (NFS) y a nivel de bloque (iSCSI).
Para NFS, se menciona la posibilidad de aplicar ACLs y cuotas por usuario desde TrueNAS.

Gestión Centralizada de Identidades con FreeIPA

Se introduce IPA Server como una solución integrada para la gestión centralizada de usuarios, grupos, autenticación (basada en Kerberos) y autorización, principalmente para sistemas Linux/Unix.
Se describe el proceso de instalación y configuración de IPA Server, la alta de usuarios y la integración con DNS y LDAP.

Plataforma de Virtualización Proxmox VE

Se aborda la instalación de Proxmox VE y su configuración básica.
Se detalla la integración de Proxmox con el servidor IPA para la autenticación de usuarios, lo que permite que los usuarios gestionados centralmente en IPA puedan acceder y utilizar recursos en Proxmox.
Se explica cómo configurar la sincronización horaria con el servidor IPA, que también actúa como NTP server.

Gestión de Máquinas Virtuales y Contenedores

Se muestran comandos básicos para listar, parar y configurar VMs. Se describe cómo migrar o clonar máquinas virtuales entre nodos de Proxmox, incluyendo un script para clonar una VM de un almacenamiento NFS a uno local.
También se menciona cómo obtener información de las VMs, como su dirección IP, utilizando el QEMU Guest Agent y la API de Proxmox.

Proxmox Clustering

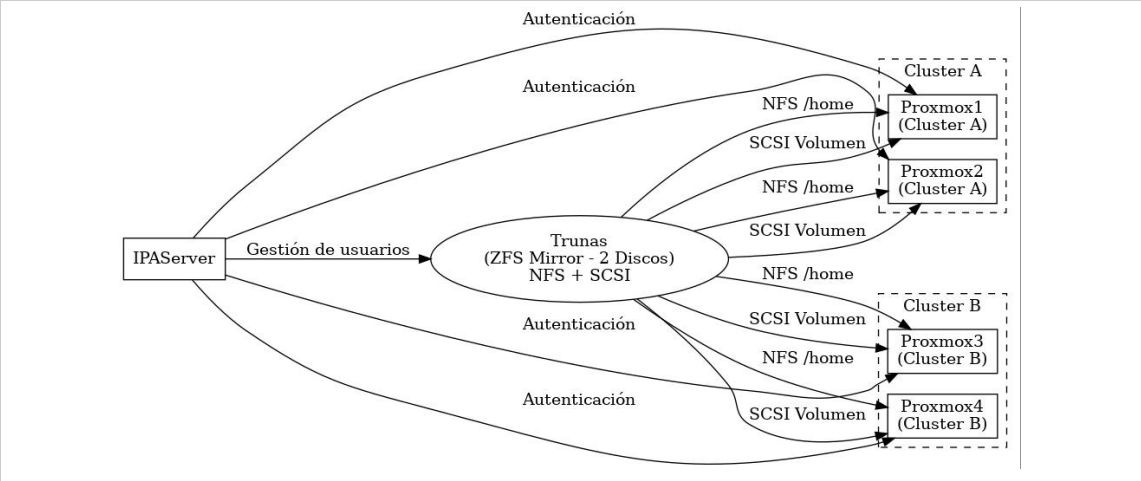
Se describe cómo configurar un clúster de Proxmox para alta disponibilidad y gestión centralizada de múltiples nodos.
Se detallan los pasos para crear un clúster, añadir nodos, solucionar problemas comunes (como VMs existentes en un nodo que intenta unirse), y eliminar nodos o el clúster completo.
Se explica el concepto de quórum y cómo verificar el estado del clúster.

Gestión de Permisos y Roles en Proxmox

Se cubre cómo otorgar permisos a usuarios en Proxmox, especialmente a aquellos autenticados a través de IPA.
Se describen roles como PVEVMUser y PVEVMAdmin y las acciones que permiten. Se explica cómo asignar permisos a nivel de máquina virtual o nodo.
Se presentan scripts y ejemplos para otorgar permisos de forma temporal.

Automatización

El documento incluye varios scripts para automatizar tareas como la instalación de clientes IPA en nodos y la gestión de máquinas virtuales y permisos.



Aplicación para la gestión de aulas de alumnos

La clave reside en la **integración de FreeIPA con Proxmox y el sistema de control de acceso (ACL) de Proxmox VE**

- **Identidades Centralizadas**:

FreeIPA permite gestionar todos los usuarios (alumnos y profesores) desde un único punto. Cada alumno tiene una cuenta única en IPA.

- **Asignación de VMs por Asignatura/Laboratorio**:

En Proxmox, las máquinas virtuales (VMs) se pueden crear y configurar para las necesidades específicas de cada asignatura o laboratorio.
Utilizando el sistema de permisos de Proxmox, se pueden asignar acceso a VMs o grupos de VMs a usuarios o grupos de IPA específicos. Por ejemplo:

- Crear un grupo de IPA por asignatura
- Crear las VMs necesarias para la asignatura de Redes.
- En Proxmox, asignar el rol 'PVEVMUser' o 'PVEVMAdmin' a nivel de esas VMs o de una carpeta que las contenga, específicamente para el grupo 'alumnos_redes@IESPOVEDA.COM' (asumiendo 'IESPOVEDA.COM' es tu realm de IPA). Esto permite que solo los miembros de ese grupo IPA (los alumnos inscritos en Redes) puedan ver y gestionar (encender, apagar, etc.) sus VMs.

- **Control de Acceso Granular**:

El sistema de roles de Proxmox (como 'PVEVMUser' que solo permite encender/apagar, o 'PVEVMAdmin' que permite más gestión) te da flexibilidad sobre lo que los alumnos pueden hacer con las VMs asignadas.

- **Automatización del Acceso**:

Los scripts de 'pveum aclmod' y 'pveum acldel' permiten automatizar la concesión y revocación de permisos.

Puedes, por ejemplo, usar scripts para habilitar automáticamente el acceso a las VMs al inicio de un laboratorio y revocarlo al finalizar, o asignar VMs a grupos de usuarios automáticamente basándote en listas.

- **Flexibilidad de Almacenamiento**:

El uso de TrueNAS con NFS permite tener un almacenamiento centralizado para las imágenes de disco de las VMs.

Esto facilita la creación de copias (clonado) de una VM base para cada alumno o grupo, o el despliegue rápido de laboratorios idénticos.

Si el rendimiento es crítico para un laboratorio, el script para mover la VM del almacenamiento NFS a uno local en el nodo de Proxmox ofrece una solución, sin perder la gestión centralizada de identidades y permisos.

Estrategia

La estrategia se basa en centralizar la identidad (IPA) y los datos de usuario (TrueNAS NFS), pero descentralizar la ejecución de las VMs y sus discos (copiándolos a almacenamiento local en cada nodo Proxmox) para sortear las limitaciones de red de un entorno sin VLANs dedicadas a almacenamiento. La automatización con scripts y cron facilita el despliegue masivo y la gestión de permisos.
La arquitectura principal involucra Proxmox VE para la virtualización, TrueNAS para el almacenamiento centralizado e IPA Server (FreeIPA) para la gestión centralizada de identidades y servicios de red.

- **Autenticación y Gestión de Usuarios:**
  - IPA Server es el núcleo para la gestión de usuarios, grupos y autenticación, utilizando principalmente Kerberos y LDAP.
  - Los nodos Proxmox se integran con el IPA Server como clientes para autenticar a los usuarios (por ejemplo, a través de LDAP o SSSD).
  - Se da de alta a los usuarios (como "arcadio") en el IPA Server.
- **Almacenamiento Centralizado (TrueNAS):**
  - TrueNAS CORE gestiona el almacenamiento mediante ZFS, ofreciendo funcionalidades como pools y datasets.
  - Se utiliza NFS para compartir almacenamiento desde TrueNAS.
  - Un uso clave de NFS en este escenario es el almacenamiento de datos de usuario, como directorios *home*, donde los datos de los alumnos residen de forma centralizada en TrueNAS. Se pueden configurar datasets específicos por usuario en TrueNAS y aplicar cuotas con `zfs set quota=`.
  - Aunque el almacenamiento centralizado (iSCSI o NFS) en TrueNAS puede usarse para imágenes de disco de máquinas virtuales, la idea para la clase es evitar que las VMs se ejecuten directamente desde este almacenamiento compartido debido a las limitaciones de red (falta de VLANs y comunicaciones lentas).
- **Proceso de Preparación y Ejecución de la Clase:**
  - Planificación y Configuración
    - El entorno base (Proxmox cluster, TrueNAS, IPA) se configura previamente, incluyendo la integración de Proxmox y TrueNAS con IPA para la autenticación y acceso a los recursos.
    - Se configura el montaje NFS de los datos de usuario (homes) desde TrueNAS en los nodos Proxmox.
    - Se sincroniza el tiempo de todos los nodos con el IPA Server NTP.
  - Preparación por el Profesor
    - El profesor se conecta a un nodo (probablemente uno con acceso a una plantilla o imagen de disco base de la VM).
    - Crea la configuración de la máquina virtual para la clase.
    - Define un archivo JSON que especifica qué VMs (basadas en IDs de origen) deben desplegarse en qué nodos de destino (`nodo_destino`) y a qué usuario (`usuario`) se le

asignará permiso en cada nodo.

- Despliegue de VMs (Local)
  - Se utiliza un script (como `migrar_migrar_varias_vms.sh`) que procesa el archivo JSON.
  - Este script invoca a otro script (como `migrar_vm_desde_nodo.sh`) en cada nodo de destino.
  - La función de este script es clonar o copiar la imagen de disco de la VM de origen a un almacenamiento *local* (`/var/lib/vz/images/`) en el nodo de destino.
  - Esto permite que la VM se ejecute con los discos en almacenamiento local, mitigando el impacto de las comunicaciones lentas.
  - El script también crea la configuración de la nueva VM localmente en el nodo de destino.
- Asignación de Permisos
  - El script de clonación (`migrar_vm_desde_nodo.sh`) incluye un paso para asignar permisos al usuario específico (`USUARIO`) sobre la VM clonada (`VMID_NUEVO`) en el nodo de destino utilizando `pveum aclmod`.
  - Se les asigna un rol como `PVEVMUser` que les permite iniciar, apagar y ver el estado de la VM, pero no modificar su hardware o configuración.
  - Estos permisos, gestionados por Proxmox, se aplican a usuarios autenticados contra el IPA Server a través de la integración configurada.
  - Los permisos asignados con `pveum aclmod` se almacenan de forma distribuida en el cluster (`pmxcfs`).
- Automatización
  - El proceso de despliegue y asignación de permisos puede automatizarse mediante tareas programadas (cron jobs) para ejecutarse en un momento dado (ej. antes de la clase).
  - También se pueden usar scripts para automatizar la revocación de permisos.

- **Finalización de la Clase:**
  - Una vez terminada la clase, se revoan los permisos que se habían concedido a los usuarios sobre las VMs. Esto se puede hacer manualmente con `pveum acldel` o automáticamente con scripts y cron jobs.
  - Los datos de los alumnos permanecen centralizados en TrueNAS vía NFS, mientras que las VMs locales y sus discos pueden ser apagados o incluso eliminados si no se necesitan para futuras clases.

IPA Server

Rocky Linux para IPA Server

Instalando Rocky linux (NO MV)

Lo mejor es **instalarlo en un servidor independiente** (Seguir los mismos pasos que a continuación pero en un servidor independiente)

Instalación Rocky linux (Como MV)

EN EL EJEMPLO UTILIZAMOS UN PROXMOX EN EL QUE INCLUIR COMO MÁQUINA VIRTUAL IPA SERVER. ESTO ES SÓLO EN LABORATORIO LUEGO NO SE HACE.
CADA SERVIDOR SE TRATA POR SEPARADO, EN MÁQUINAS FÍSICAS DISTINTAS
Pasamos nuestro ido de instalación al directorio `/var/lib/vz/template/iso/.`

```
mount /dev/sdb1 /mnt/usb
cp /mnt/usb/Rocky-9.5-x86_64-dvd.iso /var/lib/vz/template/iso/.
umount /mnt/usb
```

No sería adecuado crear un contenedor, se recomienda una MV, de todasformas si quisiéramos hacerlo podríamos de la manera:
Ejemplo: pct create -hostname -rootfs -memory -cores -net0 name=eth0,bridge=vmbr0,ip=,gw=

No se recomienda FreeIPA en un contenedor

- Limitaciones de aislamiento
- Requisitos de seguridad y red: FreeIPA gestiona autenticaciones y servicios críticos como Kerberos
- Compatibilidad de servicios
- Flexibilidad y control total

```
fdisk /dev/sda
```

Preparando disco duro

He creado el /dev/sdba4 para poder instalar el rocky linux

```
root@pm153:~# lsblk
NAME                                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
sda                                   8:0    0 931.5G  0 disk
├─sda1                                8:1    0  1007K  0 part
├─sda2                                8:2    0     1G  0 part /boot/efi
├─sda3                                8:3    0   749G  0 part
│ ├─pve-swap                        252:4    0     8G  0 lvm  [SWAP]
│ ├─pve-root                        252:5    0    96G  0 lvm  /
│ ├─pve-data_tmeta                  252:6    0   6.3G  0 lvm
│ │ └─pve-data                      252:8    0 616.4G  0 lvm
│ └─pve-data_tdata                  252:7    0 616.4G  0 lvm
│   └─pve-data                      252:8    0 616.4G  0 lvm
└─sda4
```

```
mkfs.ext4 /dev/sda4
mkdir /mnt/rocky
mount /dev/sda4 /mnt/rocky
```

```
root@pm153:~# cat /etc/fstab
...
...
#
/dev/sda4 /mnt/rocky ext4 defaults 0 2
```

```
root@pm153:~# pvs
  PV             VG               Fmt  Attr PSize    PFree
  /dev/nvme0n1p3 pve-OLD-5914B1C8 lvm2 a--  <930.51g 16.00g
  /dev/sda3      pve              lvm2 a--  <749.00g 16.00g
```

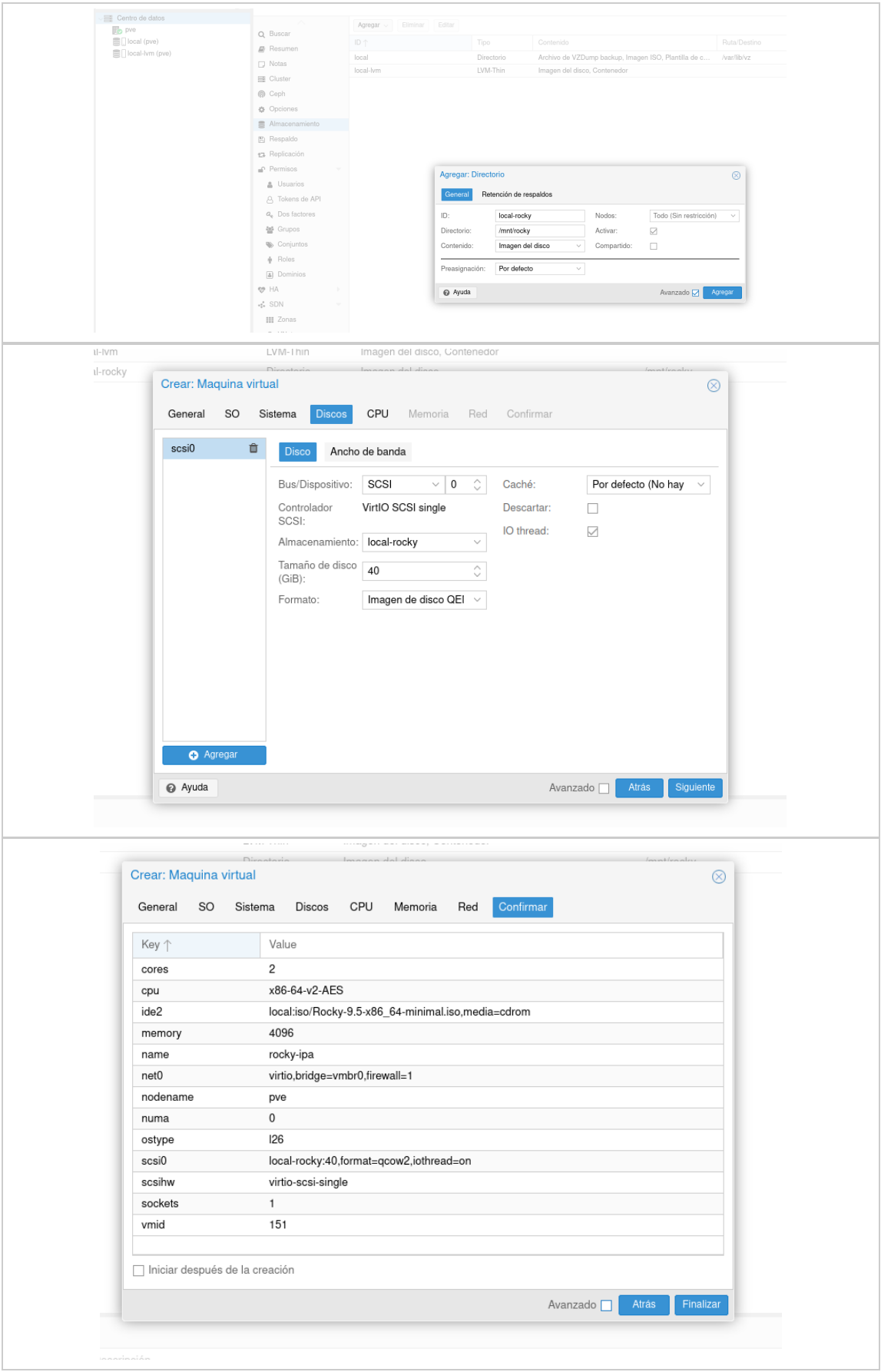
Ambos volúmenes físicos utilizan el formato lvm2, que es el formato estándar de LVM (Logical Volume Manager).

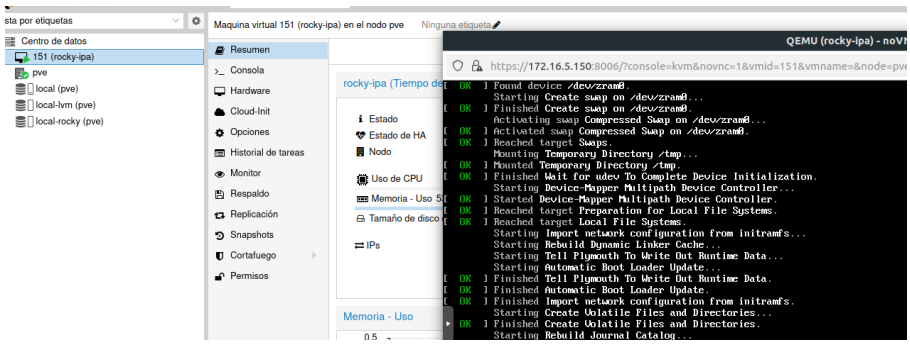
- local-lvm en Proxmox usa el grupo de volúmenes pve (el que aparece en tu salida de pvs como pve), para asignar almacenamiento dinámico a los contenedores.
- La configuración local-lvm probablemente está utilizando el grupo de volúmenes pve, ya que es el grupo de volúmenes predeterminado de Proxmox. El formato de almacenamiento en LVM es gestionado a nivel de LVM, por lo que no necesitas preocuparte por los formatos de disco como qcow2 o raw.

```
root@pm153:~# vgs
  VG               #PV #LV #SN Attr   VSize    VFree
  pve                1   3   0 wz--n- <749.00g 16.00g
  pve-OLD-5914B1C8   1   3   0 wz--n- <930.51g 16.00g
```

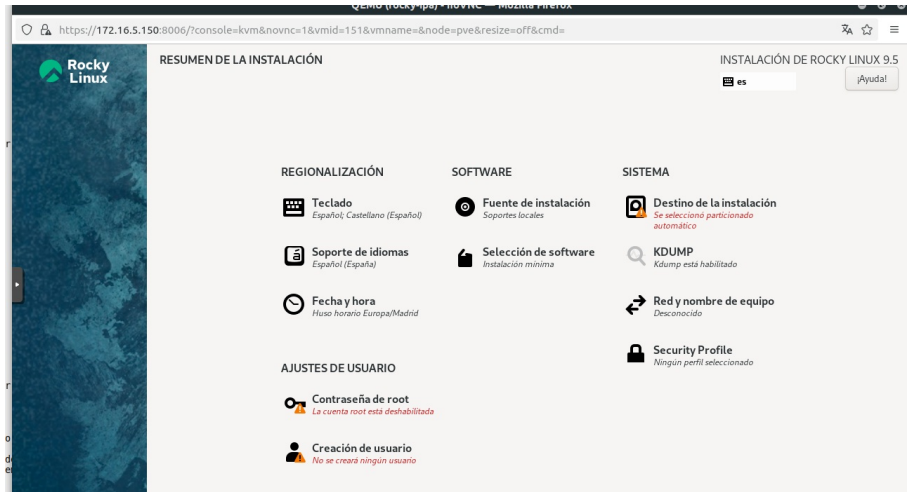
Máquina virtual

Instalamos la máquina virtual de Rocky linux

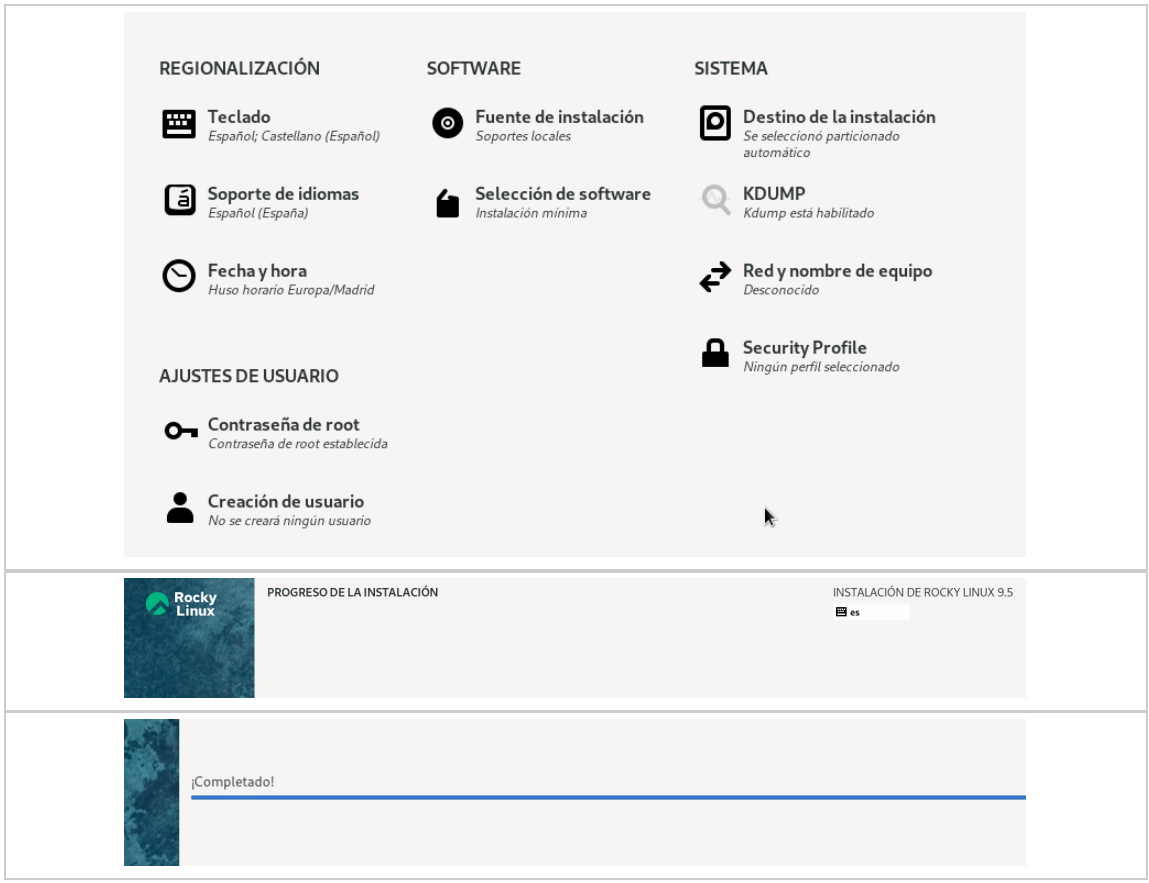



Centro de datos
151 (rocky-ipa)
pve
local (pve)
local-lvm (pve)
local-rocky (pve)
Maquina virtual 151 (rocky-ipa) en el nodo pve
Resumen
Consola
Hardware
Cloud-Init
Opciones
Historial de tareas
Monitor
Respaldo
Replicación
Snapshots
Cortafuego
Permisos
QEMU (rocky-ipa) - noVN
https://172.16.5.150:8006/?console=kvm&novnc=1&vmid=151&vmname=&node=pve
Found device /dev/zram0.
Starting Create swap on /dev/zram0...
Finished Create swap on /dev/zram0.
Activating swap Compressed Swap on /dev/zram0...
Activated swap Compressed Swap on /dev/zram0.
Reached target Swaps.
Mounting Temporary Directory /tmp...
Mounted Temporary Directory /tmp.
Finished Wait for udev To Complete Device Initialization.
Starting Device-Mapper Multipath Device Controller...
Started Device-Mapper Multipath Device Controller.
Reached target Preparation for Local File Systems.
Reached target Local File Systems.
Starting Import network configuration from initramfs...
Starting Rebuild Dynamic Linker Cache...
Starting Tell Plymouth To Write Out Runtime Data...
Starting Automatic Boot Loader Update...
Finished Tell Plymouth To Write Out Runtime Data.
Finished Automatic Boot Loader Update.
Finished Import network configuration from initramfs.
Starting Create Volatile Files and Directories...
Finished Create Volatile Files and Directories.
Starting Rebuild Journal Catalog...

Rocky Linux
INSTALACIÓN DE ROCKY LINUX
BIENVENIDO A ROCKY LINUX 9.5.
¿Qué idioma quiere utilizar durante el proceso de instalación?
English
العربية
Français
Deutsch
日本語
中文
Русский
Español
Afrikaans
Español (España)
Español (Venezuela)
Español (Uruguay)
Español (Estados Unidos)
Español (El Salvador)
Español (Paraguay)
Español (Puerto Rico)
Español (Perú)
Español (Panamá)
Español (Nicaragua)


https://172.16.5.150:8006/?console=kvm&novnc=1&vmid=151&vmname=&node=pve&resize=off&cmd=
Rocky Linux
RESUMEN DE LA INSTALACIÓN
INSTALACIÓN DE ROCKY LINUX 9.5
¡Ayuda!
REGIONALIZACIÓN
Teclado
Español; Castellano (Español)
Soporte de idiomas
Español (España)
Fecha y hora
Huso horario Europa/Madrid
SOFTWARE
Fuente de instalación
Soportes locales
Selección de software
Instalación mínima
SISTEMA
Destino de la instalación
Se seleccionó particionado automático
KDUMP
Kdump está habilitado
Red y nombre de equipo
Desconocido
Security Profile
Ningún perfil seleccionado
AJUSTES DE USUARIO
Contraseña de root
La cuenta root está deshabilitada
Creación de usuario
No se creará ningún usuario

DESTINO DE LA INSTALACIÓN
Hecho
Selección de dispositivo
Seleccione los dispositivos en que le gustaría instalar. Se mantendrán sin tocar hasta que pulse el botón «Comenzar instalación» del menú princi
Discos estándares locales
40 GiB
QEMU QEMU HARDDISK
sda / 40 GiB libre
Discos especializados y de red
Añadir un disco...
Configuración de almacenamiento
Automática
Personalizada



IP estática

Pra ver los device posibles:

```
nmcli device
DEVICE  TYPE      STATE                   CONNECTION
ens18   ethernet  connected               ens18
lo      loopback  connected (externally)  lo
```

```
nmcli con mod ens18 ipv4.addresses 172.16.5.110/16
nmcli con mod ens18 ipv4.gateway 172.16.0.1
nmcli con mod ens18 ipv4.dns "8.8.8.8 8.8.4.4"
nmcli con mod ens18 ipv4.method manual
```

Verificamos:

```
ip a
1: lo: <loopback,up,lower_up> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: ens18: <broadcast,multicast,up,lower_up> mtu 1500 qdisc fq_codel state UP group default
qlen 1000
    link/ether bc:24:11:1b:55:25 brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    inet 172.16.5.110/16 brd 172.16.255.255 scope global noprefixroute ens18
       valid_lft forever preferred_lft forever
    inet6 fe80::be24:11ff:fe1b:5525/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
</broadcast,multicast,up,lower_up></loopback,up,lower_up>
```

IPA sever

Introducción

IPA Server (Identity, Policy, and Audit Server) es una solución de gestión de identidades basada en FreeIPA, que es un proyecto de código abierto. IPA Server proporciona un conjunto completo de servicios relacionados con la autenticación, autorización y auditoría, todo en un único sistema integrado. Está diseñado principalmente para gestionar redes de Linux y sistemas basados en Unix.

Principales **características** de IPA Server:

Gestión centralizada de identidades

IPA Server permite administrar usuarios, grupos, contraseñas y otras identidades de manera centralizada.

Esto significa que puedes gestionar todas las cuentas de usuario y sus permisos desde un único lugar.

Autenticación y autorización

IPA Server integra la autenticación basada en Kerberos, lo que permite a los usuarios autenticarse de manera segura en la red sin necesidad de enviar contraseñas a través de la red.

Además, gestiona las políticas de acceso y autorización a través de la red.

Integración con LDAP

Utiliza LDAP (Lightweight Directory Access Protocol) para almacenar los datos relacionados con las identidades, como la información de usuarios, grupos, dispositivos, etc.

Autenticación multi-protocolo

IPA Server soporta varios protocolos de autenticación, como Kerberos, LDAP, DNS, y NTP.

Ofrece soporte para la autenticación de usuarios y equipos en la red, lo que facilita la gestión de la infraestructura de red.

Políticas de seguridad

Permite implementar políticas de seguridad, como la expiración de contraseñas, complejidad de contraseñas y otras reglas relacionadas con la seguridad de las identidades.

Auditoría y registro

IPA Server incluye capacidades de auditoría que permiten registrar eventos importantes en el sistema, como inicios de sesión, cambios en las políticas y modificaciones de cuentas, lo que facilita el seguimiento y la investigación de incidentes de seguridad.

Integración con DNS

IPA Server puede actuar como servidor de DNS para proporcionar resolución de nombres en la red, lo que simplifica la gestión de direcciones y servicios dentro de un dominio.

Interfaz web de administración

IPA Server tiene una interfaz web que permite a los administradores gestionar el sistema de forma sencilla y visual, sin necesidad de estar constantemente utilizando la línea de comandos.

Componentes principales de IPA Server:

- **Kerberos**: Sistema de autenticación basado en tickets que permite a los usuarios y equipos autenticarse de manera segura.
- **LDAP**: Protocolo utilizado para almacenar y acceder a la información sobre usuarios, grupos y equipos en una red.
- **DNS**: Servidor de nombres de dominio, utilizado para resolver nombres dentro de la red.
- **NTP**: Protocolo para la sincronización del tiempo en la red.
- **Dogtag**: Sistema de gestión de certificados para emitir y gestionar certificados digitales en la red.

Casos de uso de IPA Server:

- **Gestión de usuarios y dispositivos**: Centraliza la gestión de usuarios y dispositivos en una red.
- **Autenticación única (SSO)**: Permite la autenticación única para múltiples servicios y sistemas en la red.
- **Control de acceso**: Define políticas de acceso y permisos para distintos usuarios y grupos dentro de la red.
- **Sistemas de seguridad**: Mejora la seguridad mediante el uso de Kerberos, autenticación multi-factor, políticas de contraseñas, etc.

INSTALACIÓN

Actualizamos el sistema:

```
sudo dnf update -y
```

Instalamos los paquetes necesarios pra el ipa-server:

- ipa-server: Paquete principal para el servidor FreeIPA.
- ipa-server-dns: Paquete adicional si planeas usar DNS administrado por IPA.

Configuramos el nombre de nuestro servivor: ipa.iespoveda.com

```
/etc/hostname
ipa.iespoveda.com
```

```
/etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
#
172.16.5.110 ipa.iespoveda.com ipa
#
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
```

Si tengo activado el firewall: `sudo firewall-cmd --state`

```
sudo firewall-cmd --state
running
```

Si tienes el firewall habilitado, es necesario permitir el acceso a los puertos utilizados por el servicio IPA.

```
# Abrir puertos TCP y UDP necesarios para IPA
firewall-cmd --permanent --add-port=80/tcp
firewall-cmd --permanent --add-port=443/tcp
firewall-cmd --permanent --add-port=389/tcp
firewall-cmd --permanent --add-port=636/tcp
firewall-cmd --permanent --add-port=88/tcp
firewall-cmd --permanent --add-port=464/tcp
firewall-cmd --permanent --add-port=53/tcp
firewall-cmd --permanent --add-port=88/udp
firewall-cmd --permanent --add-port=464/udp
firewall-cmd --permanent --add-port=53/udp
firewall-cmd --permanent --add-port=123/udp

# Recargar el firewall para aplicar los cambios
firewall-cmd --reload
```

Ahora reiniciamos:

```
sudo firewall-cmd --reload
```

Verificamos con:

```
#  firewall-cmd --zone=public --list-ports
53/tcp 80/tcp 88/tcp 389/tcp 443/tcp 636/tcp 53/udp 88/udp 123/udp 389/udp
```

```
sudo dnf install ipa-server ipa-server-dns -y
```

Ahora ejecutamos:

```
sudo ipa-server-install
```

Opciones que nos va a pedir:

- Domínio del servidor IPA: El nombre del dominio de tu red, iespoveda.com.
- Nombre del servidor: El FQDN del servidor IPA (ej. ipa.iespoveda.com).
- Contraseña del administrador: Contraseña para el administrador de IPA (usuario admin).
- Configuración de DNS (opcional): Si queremos usar el servidor DNS de IPA, se te pedirá que lo configures durante el proceso.

Passwords que nos va a pedir:

- Directory Manager password

Certain directory server operations require an administrative user. This user is referred to as the Directory Manager and has full access to the Directory for system management tasks and will be added to the instance of directory server created for IPA. The password must be at least 8 characters long.

- IPA admin password

The IPA server requires an administrative user, named 'admin'. This user is a regular system account used for IPA server administration.

```
...
...
...
The IPA Master Server will be configured with:
Hostname:       ipa.iespoveda.com
IP address(es): 172.16.5.110
Domain name:    iespoveda.com
Realm name:     IESPOVEDA.COM

The CA will be configured with:
Subject DN:   CN=Certificate Authority,O=IESPOVEDA.COM
Subject base: O=IESPOVEDA.COM
Chaining:     self-signed

BIND DNS server will be configured to serve IPA domain with:
Forwarders:       1.1.1.1, 8.8.8.8
Forward policy:   only
Reverse zone(s):  5.16.172.in-addr.arpa.

NTP server: hora.cica.es
NTP server: hora.rediris.es
Continue to configure the system with these values? [no]: yes
...
...
...
```

y como final:

```
==============================================================================
Setup complete

Next steps:
 1. You must make sure these network ports are open:
  TCP Ports:
    * 80, 443: HTTP/HTTPS
    * 389, 636: LDAP/LDAPS
    * 88, 464: kerberos
    * 53: bind
  UDP Ports:
    * 88, 464: kerberos
    * 53: bind
    * 123: ntp

 2. You can now obtain a kerberos ticket using the command: 'kinit admin'
    This ticket will allow you to use the IPA tools (e.g., ipa user-add)
    and the web user interface.

Be sure to back up the CA certificates stored in /root/cacert.p12
These files are required to create replicas. The password for these
files is the Directory Manager password
The ipa-server-install command was successful
```

```
systemctl status ipa
● ipa.service - Identity, Policy, Audit
     Loaded: loaded (/usr/lib/systemd/system/ipa.service; enabled; preset: disabled)
     Active: active (exited) since Fri 2025-03-14 15:12:19 CET; 2min 58s ago
    Process: 15627 ExecStart=/usr/sbin/ipactl start (code=exited, status=0/SUCCESS)
   Main PID: 15627 (code=exited, status=0/SUCCESS)
        CPU: 776ms

Mar 14 15:12:19 ipa.iespoveda.com ipactl[15627]: Starting Directory Service
Mar 14 15:12:19 ipa.iespoveda.com ipactl[15627]: Starting krb5kdc Service
Mar 14 15:12:19 ipa.iespoveda.com ipactl[15627]: Starting kadmin Service
Mar 14 15:12:19 ipa.iespoveda.com ipactl[15627]: Starting named Service
Mar 14 15:12:19 ipa.iespoveda.com ipactl[15627]: Starting httpd Service
Mar 14 15:12:19 ipa.iespoveda.com ipactl[15627]: Starting ipa-custodia Service
Mar 14 15:12:19 ipa.iespoveda.com ipactl[15627]: Starting pki-tomcatd Service
Mar 14 15:12:19 ipa.iespoveda.com ipactl[15627]: Starting ipa-otpd Service
Mar 14 15:12:19 ipa.iespoveda.com ipactl[15627]: Starting ipa-dnskeysyncd Service
Mar 14 15:12:19 ipa.iespoveda.com systemd[1]: Finished Identity, Policy, Audit.
```

Si no estuviera activo echamos a andar el servicio:

```
systemctl start ipa
systemctl status ipa
```

```
[root@ipa ~]#  ipactl status
Directory Service: RUNNING
krb5kdc Service: RUNNING
kadmin Service: RUNNING
named Service: RUNNING
httpd Service: RUNNING
ipa-custodia Service: RUNNING
pki-tomcatd Service: RUNNING
ipa-otpd Service: RUNNING
ipa-dnskeysyncd Service: RUNNING
ipa: INFO: The ipactl command was successful
```

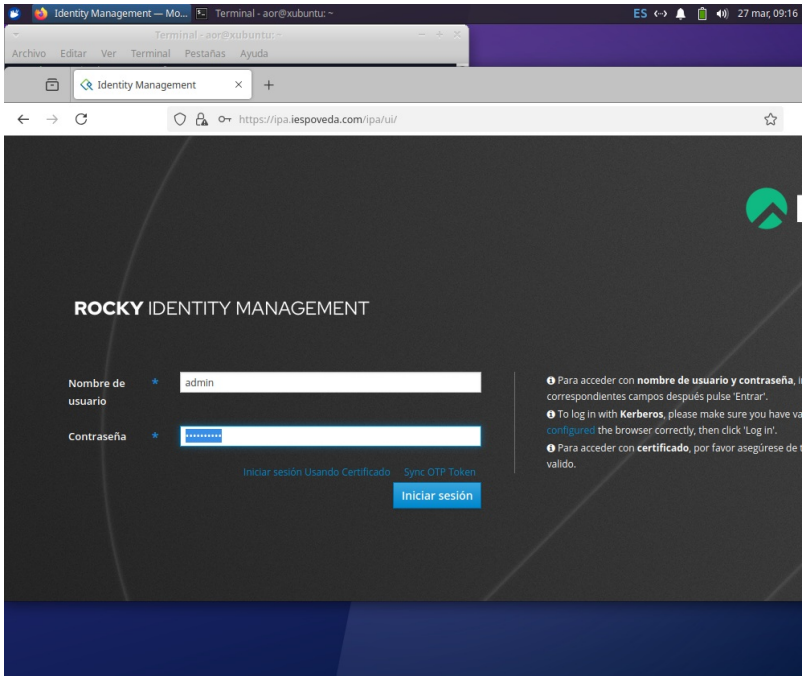
Ahora nos conectamos a:

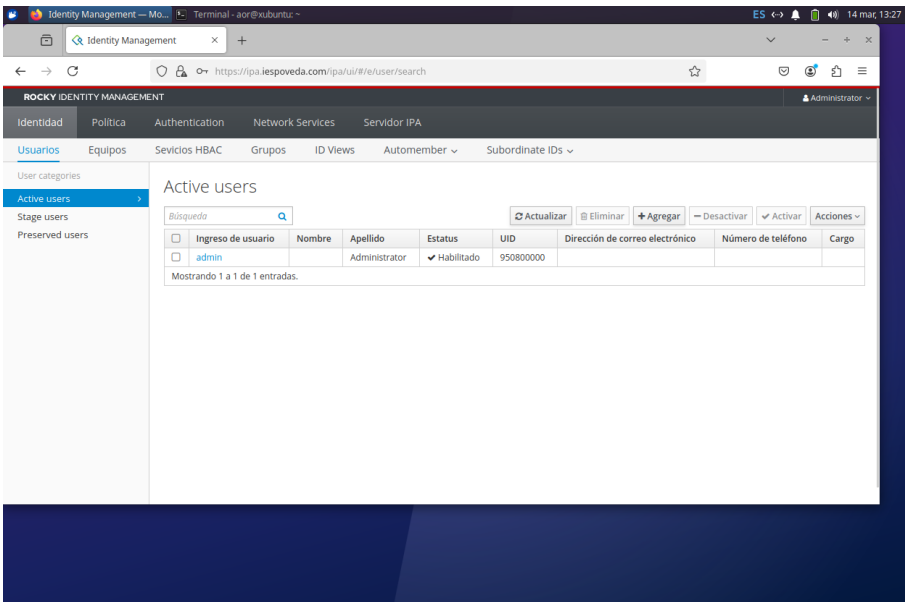
```
https://ipa.iespoveda.com/ipa/ui
```

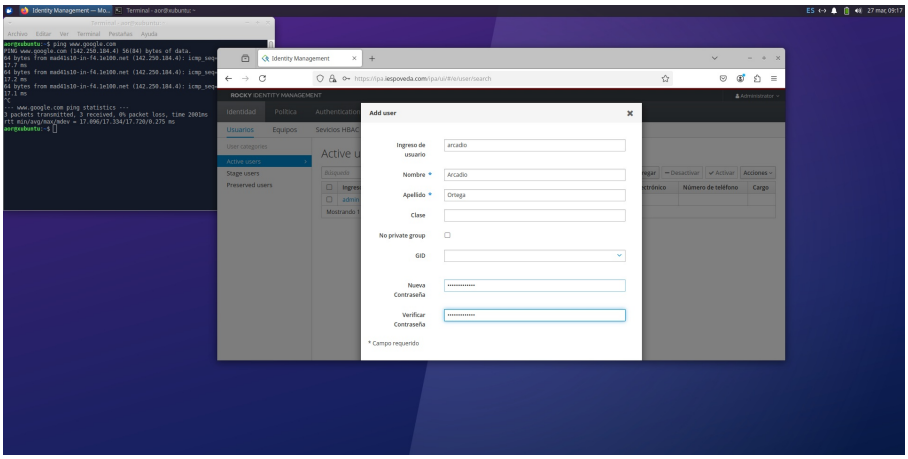
Entramos con el usuario **admin** y el password que le hayamos dado en instalación.

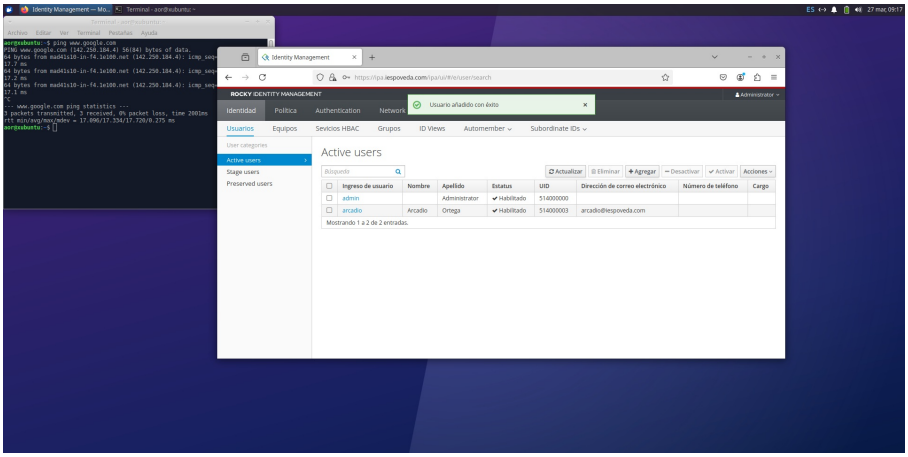
Alta de usuario

Es importante dar de alta un usuario para luego probar la integración enter proxmox e ipa. He dado de usuario: arcadio




ROCKY IDENTITY MANAGEMENT
Identidad
Política
Authentication
Network Services
Servidor IPA
Usuarios
Equipos
Sevicios HBAC
Grupos
ID Views
Automember
Subordinate IDs
User categories
Active users
Stage users
Preserved users
Active users
Actualizar
Eliminar
Agregar
Desactivar
Activar
Acciones
Ingreso de usuario
Nombre
Apellido
Estatus
UID
Dirección de correo electrónico
Número de teléfono
Cargo
admin
Administrator
Habilitado
950800000
Mostrando 1 a 1 de 1 entradas.


Add user
Ingreso de usuario
arcadio
Nombre
Arcadio
Apellido
Ortega
Clase
No private group
GID
Nueva Contraseña
Verificar Contraseña
* Campo requerido


Usuario añadido con éxito
Active users
admin
arcadio
Mostrando 1 a 2 de 2 entradas.

Proxmox

Instalación PROXMOX

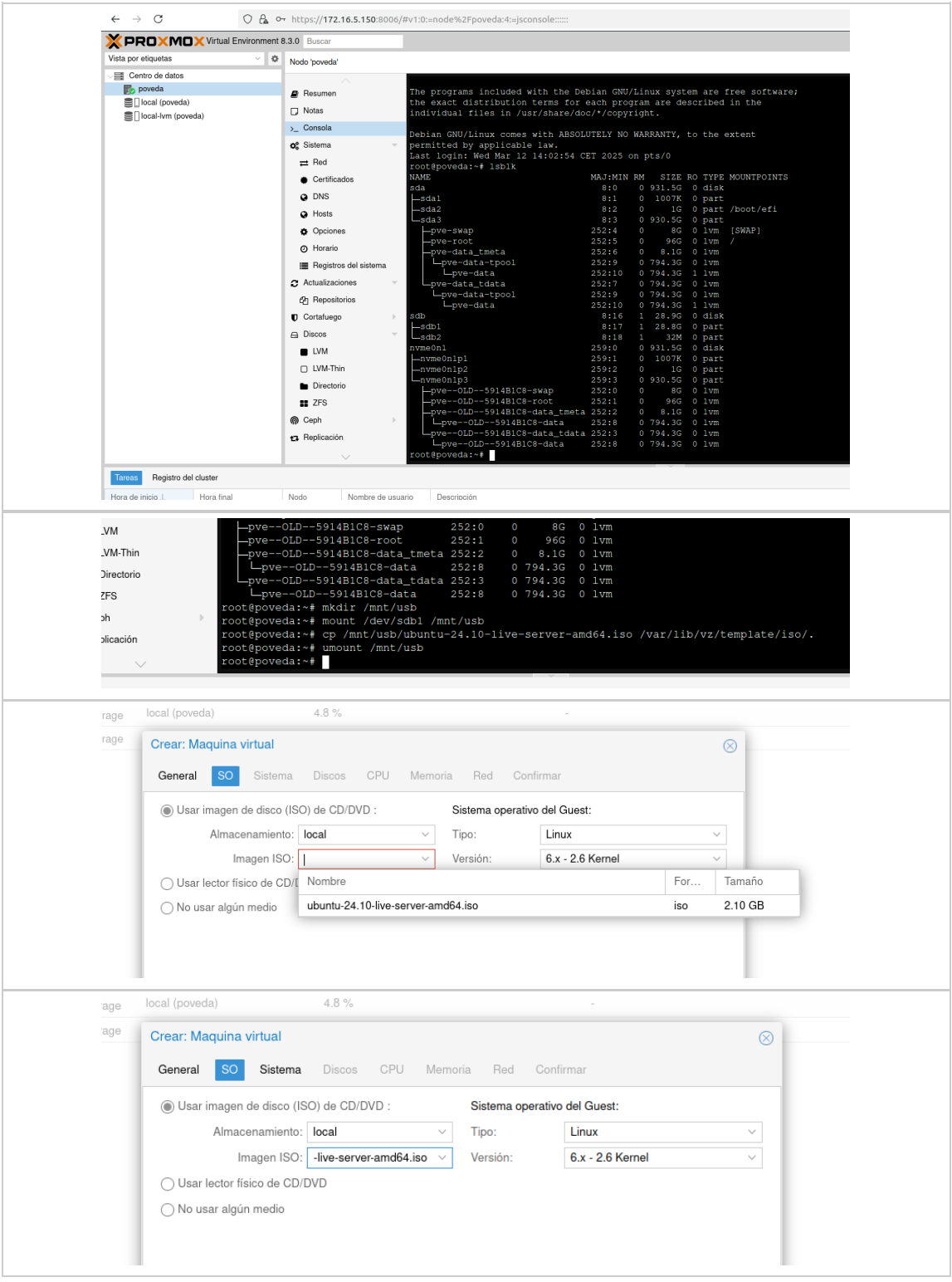
Tenemos 2 dscos duro:`sda` y `nvme0n1` Cada uno de 1TB
Vamos a instalar proxmox en el disco /dev/sda y vamos a utilizar para ello 750GB
Tras la instalación quedará:

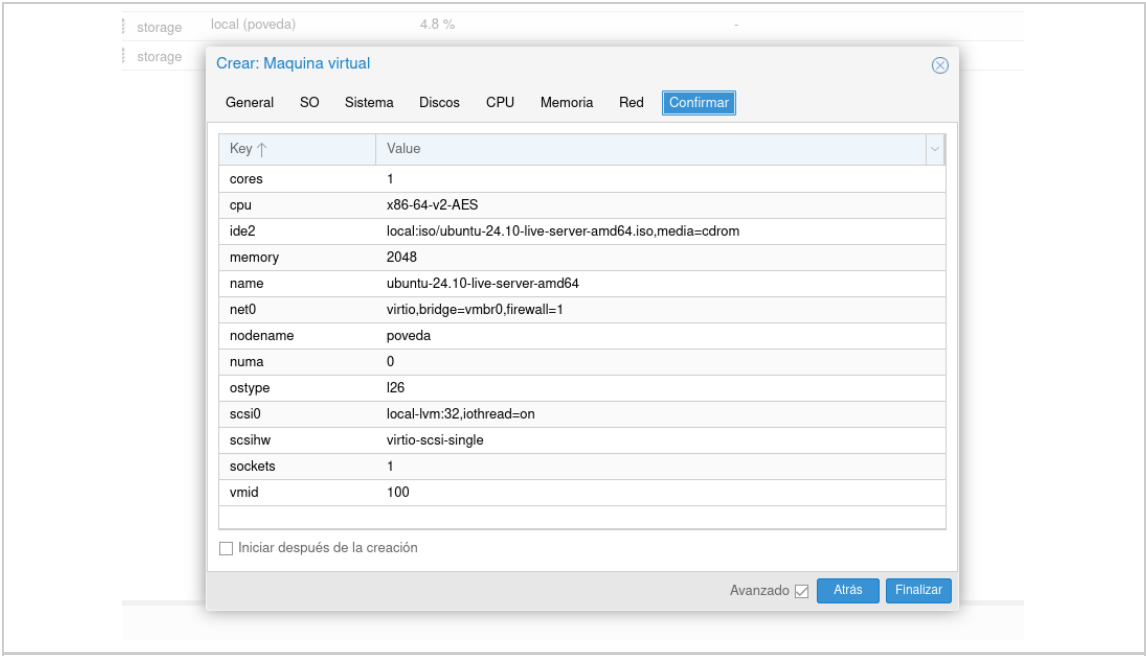
```
root@pm153:~# lsblk
NAME                         MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
sda                            8:0    0 931.5G  0 disk
├─sda1                         8:1    0  1007K  0 part
├─sda2                         8:2    0     1G  0 part /boot/efi
├─sda3                         8:3    0   749G  0 part
  ├─pve-swap                 252:4    0     8G  0 lvm  [SWAP]
  ├─pve-root                 252:5    0    96G  0 lvm  /
  ├─pve-data_tmeta           252:6    0   6.3G  0 lvm
  │ └─pve-data               252:8    0 616.4G  0 lvm
  └─pve-data_tdata           252:7    0 616.4G  0 lvm
    └─pve-data               252:8    0 616.4G  0 lvm
```

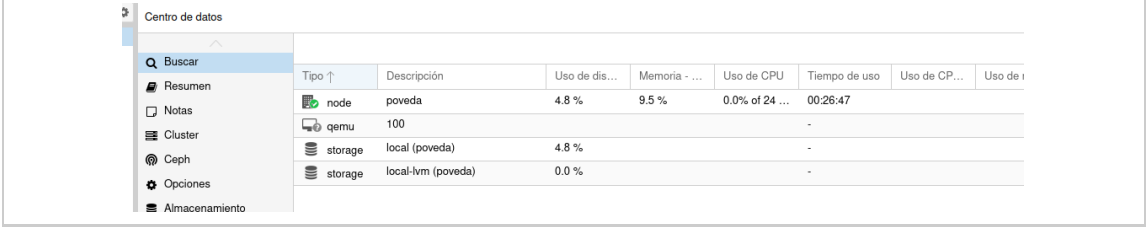
Habrá que dar entre otros:

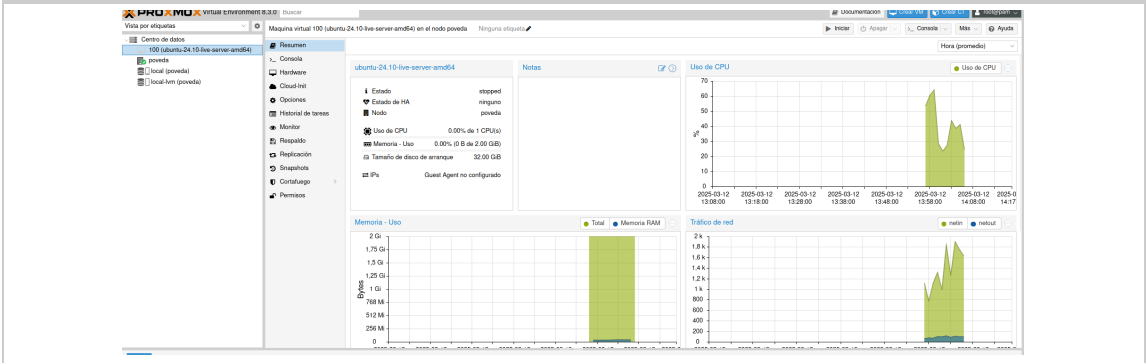
- País: Spain. Horaria: españa/Madrid
- La dirección ip. Ejemplo: 172.16.5.153
- fqdn: por ejemplo: pm153.iespoveda.com
- DNS server: 172.16.5.110 . Nuestro servidor **IPA server**

PROXMOX. Ejemplo máquina virtual




storage local (poveda) 4.8 %
storage
Crear: Maquina virtual
General SO Sistema Discos CPU Memoria Red Confirmar
Key Value
cores 1
cpu x86-64-v2-AES
ide2 local:iso/ubuntu-24.10-live-server-amd64.iso,media=cdrom
memory 2048
name ubuntu-24.10-live-server-amd64
net0 virtio,bridge=vmbr0,firewall=1
nodename poveda
numa 0
ostype l26
scsi0 local-lvm:32,iothread=on
scsihw virtio-scsi-single
sockets 1
vmid 100
Iniciar después de la creación
Avanzado
Atrás
Finalizar


Centro de datos
Buscar
Resumen
Notas
Cluster
Ceph
Opciones
Almacenamiento
Tipo Descripción Uso de dis... Memoria - ... Uso de CPU Tiempo de uso Uso de CP...
node poveda 4.8 % 9.5 % 0.0% of 24 ... 00:26:47
qemu 100 -
storage local (poveda) 4.8 % -
storage local-lvm (poveda) 0.0 % -



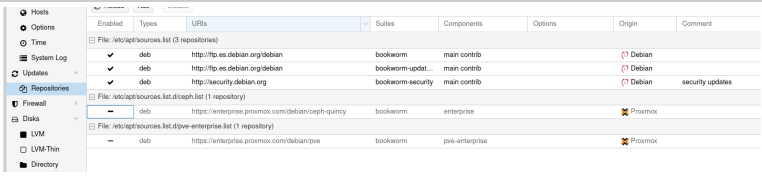
GNU GRUB version 2.12
Try or Install Ubuntu Server
Test memory

Errores de update

Tengo los siguientes errores al tratar de actualizar:

```
root@poveda:~# apt update
Hit:1 http://security.debian.org bookworm-security InRelease
Hit:2 http://ftp.es.debian.org/debian bookworm InRelease
Hit:3 http://ftp.es.debian.org/debian bookworm-updates InRelease
Err:4 https://enterprise.proxmox.com/debian/ceph-quincy bookworm InRelease
  401  Unauthorized [IP: 51.91.38.34 443]
Err:5 https://enterprise.proxmox.com/debian/pve bookworm InRelease
  401  Unauthorized [IP: 51.91.38.34 443]
Reading package lists... Done
E: Failed to fetch https://enterprise.proxmox.com/debian/ceph-quincy/dists/bookworm/InRelease
401  Unauthorized [IP: 51.91.38.34 443]
E: The repository 'https://enterprise.proxmox.com/debian/ceph-quincy bookworm InRelease' is
not signed.
N: Updating from such a repository can't be done securely, and is therefore disabled by
default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
E: Failed to fetch https://enterprise.proxmox.com/debian/pve/dists/bookworm/InRelease  401
Unauthorized [IP: 51.91.38.34 443]
E: The repository 'https://enterprise.proxmox.com/debian/pve bookworm InRelease' is not
signed.
N: Updating from such a repository can't be done securely, and is therefore disabled by
default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
```

Proxmox está intentando acceder a los repositorios de Proxmox Enterprise para actualizaciones: **401 Unauthorized**



También desde la consola:

```
root@poveda:~# grep -r "enterprise.proxmox.com" /etc/apt/sources.list.d/
/etc/apt/sources.list.d/ceph.list:deb https://enterprise.proxmox.com/debian/ceph-quincy
bookworm enterprise
/etc/apt/sources.list.d/pve-enterprise.list:deb https://enterprise.proxmox.com/debian/pve
bookworm pve-enterprise
```

```
root@poveda:~# vi /etc/apt/sources.list.d/pve-enterprise.list
#deb https://enterprise.proxmox.com/debian/pve bookworm pve-enterprise
```

Lo mismo hacemos con: `/etc/apt/sources.list.d/ceph.list` Y ahora agregamos el Proxmox no comercial

```
root@poveda:~# cat /etc/apt/sources.list.d/pve-no-subscription.list
deb http://download.proxmox.com/debian/pve bookworm pve-no-subscription
```

Ahora ya lo tenemos:

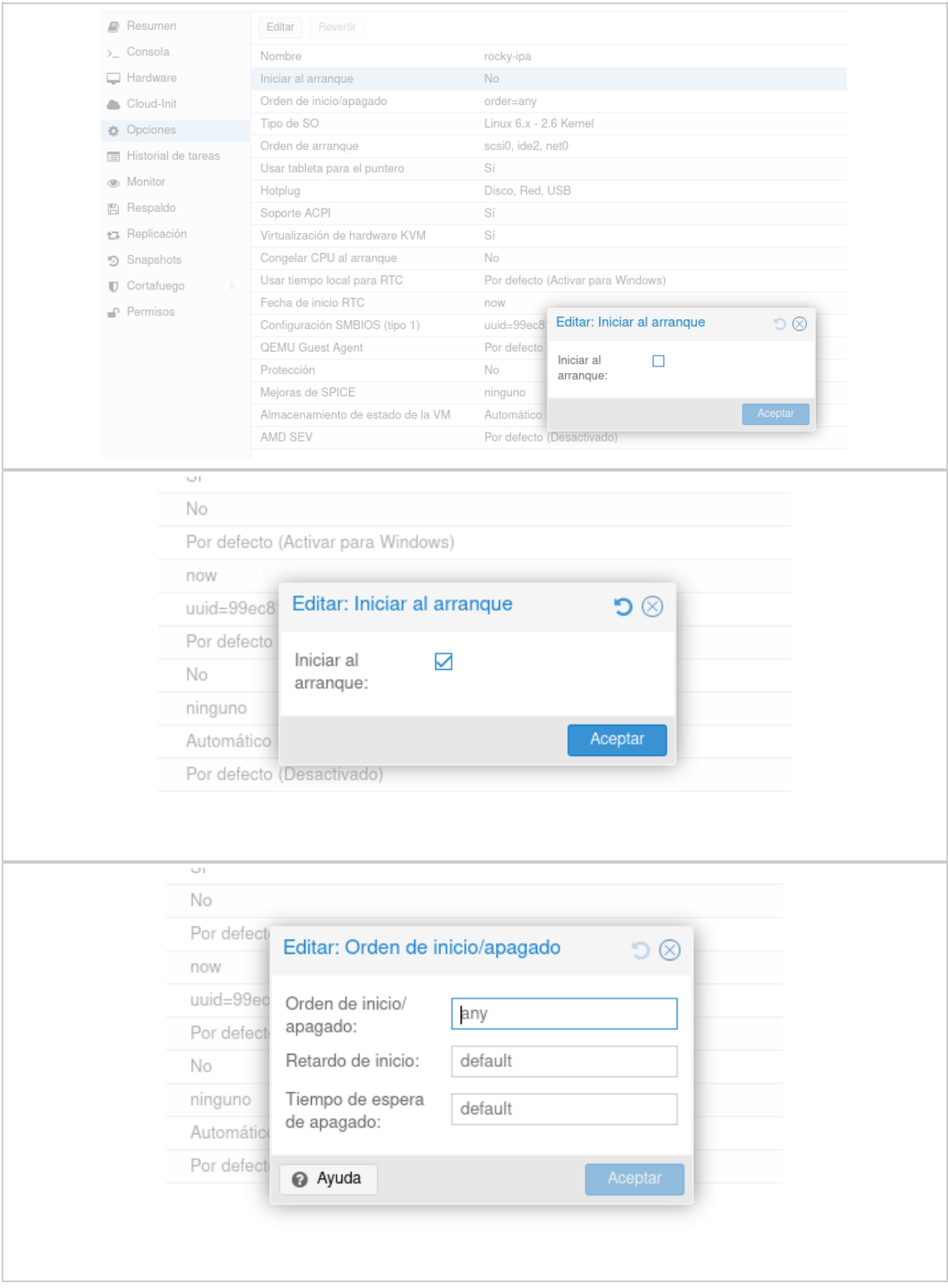
```
root@poveda:~# apt update
Hit:1 http://security.debian.org bookworm-security InRelease
Hit:2 http://ftp.es.debian.org/debian bookworm InRelease
Hit:3 http://ftp.es.debian.org/debian bookworm-updates InRelease
Hit:4 http://download.proxmox.com/debian/pve bookworm InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
87 packages can be upgraded. Run 'apt list --upgradable' to see them.
apt upgrade
....
```

Proxmox - IPA

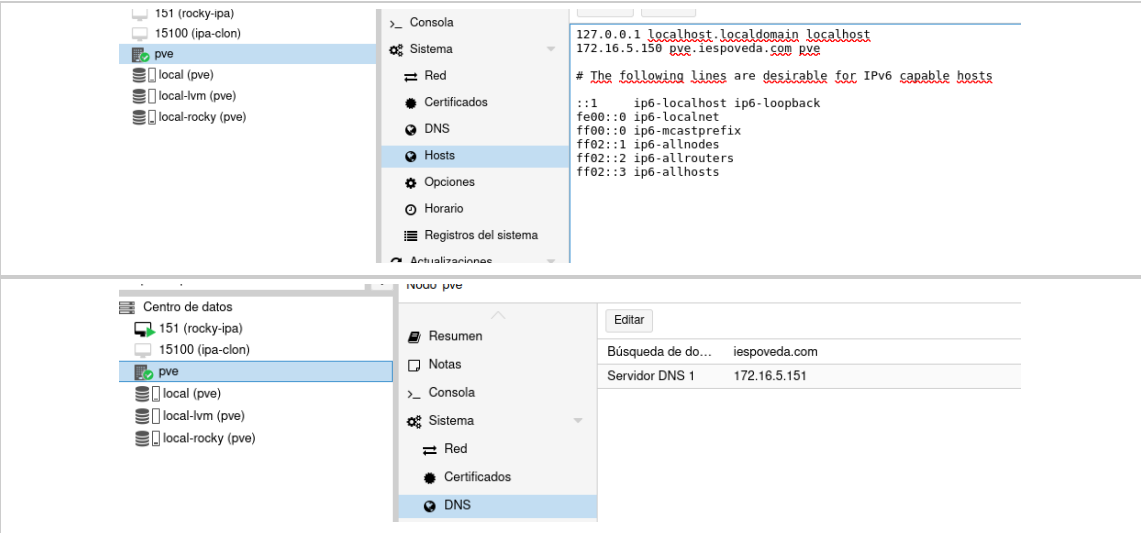
Integrar proxmox y IPA

Si está en máquina virtual (NO ES LO RECOMENDADO)

Si está instalado como una máquina virtual en proxmox necesitamos que sea la primera en arrancar **antes que proxmox autentifique usuarios**



Verificamos que proxmox esté utilizando el DNS server de nuestro IPA server:



Integramos con proxmox

Tenemos previamente lo DNS bien configurados de tal manera que:

```
dig ipa.iespoveda.com

; <<>> DiG 9.18.33-1~deb12u2-Debian <<>> ipa.iespoveda.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63894
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: e051606dabd536dc0100000067d7f1c35da08b79f7ddf992 (good)
;; QUESTION SECTION:
;ipa.iespoveda.com.  IN A

;; ANSWER SECTION:
ipa.iespoveda.com. 1200 IN A 172.16.5.110

;; Query time: 1 msec
;; SERVER: 172.16.5.110#53(172.16.5.110) (UDP)
;; WHEN: Mon Mar 17 10:56:30 CET 2025
;; MSG SIZE  rcvd: 90
```

En el servidor IPA

```
[root@ipa ~]# kinit admin
Password for admin@IESPOVEDA.COM:
[root@ipa ~]# systemctl status ipa
● ipa.service - Identity, Policy, Audit
     Loaded: loaded (/usr/lib/systemd/system/ipa.service; enabled; preset: disabled)
     Active: active (exited) since Wed 2025-03-26 14:20:12 CET; 49min ago
    Process: 8266 ExecStart=/usr/sbin/ipactl start (code=exited, status=0/SUCCESS)
   Main PID: 8266 (code=exited, status=0/SUCCESS)
        CPU: 711ms

Mar 26 14:20:12 ipa.iespoveda.com ipactl[8266]: Starting Directory Service
Mar 26 14:20:12 ipa.iespoveda.com ipactl[8266]: Starting krb5kdc Service
Mar 26 14:20:12 ipa.iespoveda.com ipactl[8266]: Starting kadmin Service
Mar 26 14:20:12 ipa.iespoveda.com ipactl[8266]: Starting named Service
Mar 26 14:20:12 ipa.iespoveda.com ipactl[8266]: Starting httpd Service
Mar 26 14:20:12 ipa.iespoveda.com ipactl[8266]: Starting ipa-custodia Service
Mar 26 14:20:12 ipa.iespoveda.com ipactl[8266]: Starting pki-tomcatd Service
Mar 26 14:20:12 ipa.iespoveda.com ipactl[8266]: Starting ipa-otpd Service
Mar 26 14:20:12 ipa.iespoveda.com ipactl[8266]: Starting ipa-dnskeysyncd Service
Mar 26 14:20:12 ipa.iespoveda.com systemd[1]: Finished Identity, Policy, Audit.
```

- Añadimos etrada dns para nuestro servidor proxmox

```
[root@ipa ~]# ipa dnsrecord-add iespoveda.com pm153 --a-ip-address=172.16.5.153
  Record name: pm153
  A record: 172.16.5.153
```

- Añadimos entrada para nuestro servidor en el ipa server

```
  [root@ipa ~]# ipa host-add pm153.iespoveda.com
--------------------------------
Added host "pm153.iespoveda.com"
--------------------------------
  Host name: pm153.iespoveda.com
  Principal name: host/pm153.iespoveda.com@IESPOVEDA.COM
  Principal alias: host/pm153.iespoveda.com@IESPOVEDA.COM
  Password: False
  Keytab: False
  Managed by: pm153.iespoveda.com
```

```
[root@ipa ~]# ipa-getkeytab -p host/pm153.iespoveda.com -k /etc/krb5.keytab
Keytab successfully retrieved and stored in: /etc/krb5.keytab
```

En el servidor proxmox pm153

```
mkdir -p /etc/ipa
chown root:root /etc/ipa
chmod 755 /etc/ipa
wget -O /etc/ipa/ca.crt http://ipa.iespoveda.com/ipa/config/ca.crt
touch /etc/krb5.conf
```

```
vi /etc/krb5.conf
[libdefaults]
    default_realm = IESPOVEDA.COM
    dns_lookup_realm = false
    dns_lookup_kdc = true

[realms]
    IESPOVEDA.COM = {
        kdc = ipa.iespoveda.com
        admin_server = ipa.iespoveda.com
    }

[domain_realm]
    .iespoveda.com = IESPOVEDA.COM
    iespoveda.com = IESPOVEDA.COM
```

Ahora en el servidor proxmox (pm153.iespoveda.com) instalamos:

```
apt install freeipa-client
```

y hacemos:

```
root@pm153:~# touch /etc/ipa/default.conf
root@pm153:~# cat /etc/ipa/default.conf
[global]
  domain = iespoveda.com
  hostname = pm153.iespoveda.com
  realm = IESPOVEDA.COM
  ipaclient = True
  server = ipa.iespoveda.com
  ca_cert = /etc/ipa/ca.crt
```

También nos hace falta:

```
apt-get install krb5-kdc
```

Ahora podemos hacer:

```
root@pm153:~# kdb5_util create -s
Initializing database '/var/lib/krb5kdc/principal' for realm 'IESPOVEDA.COM',
master key name 'K/M@IESPOVEDA.COM'
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
Enter KDC database master key:
Re-enter KDC database master key to verify:
oot@pve:~# systemctl start krb5-kdc
root@pm153:~# systemctl status krb5-kdc
krb5-kdc.service - Kerberos 5 Key Distribution Center
     Loaded: loaded (/lib/systemd/system/krb5-kdc.service; enabled; preset: enabled)
     Active: active (running) since Wed 2025-03-26 13:02:50 CET; 5s ago
    Process: 33036 ExecStart=/usr/sbin/krb5kdc -P /var/run/krb5-kdc.pid $DAEMON_ARGS
(code=exited, status=0/SUCCESS)
   Main PID: 33038 (krb5kdc)
      Tasks: 1 (limit: 18345)
     Memory: 1.2M
        CPU: 23ms
     CGroup: /system.slice/krb5-kdc.service
             └─33038 /usr/sbin/krb5kdc -P /var/run/krb5-kdc.pid

Mar 26 13:02:50 pm153 krb5kdc[33036]: Setting pktinfo on socket 0.0.0.0.88
Mar 26 13:02:50 pm153 krb5kdc[33036]: Setting up UDP socket for address ::.88
Mar 26 13:02:50 pm153 krb5kdc[33036]: setsockopt(11,IPV6_V6ONLY,1) worked
Mar 26 13:02:50 pm153 krb5kdc[33036]: Setting pktinfo on socket ::.88
Mar 26 13:02:50 pm153 krb5kdc[33036]: Setting up TCP socket for address 0.0.0.0.88
Mar 26 13:02:50 pm153 krb5kdc[33036]: Setting up TCP socket for address ::.88
Mar 26 13:02:50 pm153 krb5kdc[33036]: setsockopt(13,IPV6_V6ONLY,1) worked
Mar 26 13:02:50 pm153 krb5kdc[33036]: set up 6 sockets
Mar 26 13:02:50 pm153 systemd[1]: Started krb5-kdc.service - Kerberos 5 Key Distribution
Center.
```

Obtenemos los certificados del servidor y los ponemos direcatmente en el cliente:

```
root@pm153:~# echo | openssl s_client -showcerts -connect ipa.iespoveda.com:443 2>/dev/null |
openssl x509 -outform PEM > ipa_cert.pem
root@pm153:~# cp ipa_cert.pem /usr/local/share/ca-certificates/ipa_cert.crt
root@pm153:~# update-ca-certificates
Updating certificates in /etc/ssl/certs...
rehash: warning: skipping ca-certificates.crt,it does not contain exactly one certificate or
CRL
1 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
```

Ahora en el IPA SERVER:

```
[root@ipa ~]# ipa-getkeytab -p host/pm153.iespoveda.com -k /tmp/pm153.keytab
Keytab successfully retrieved and stored in: /tmp/pve.keytab
[root@ipa ~]# scp /tmp/pm153.keytab root@pm153.iespoveda.com:/etc/krb5.keytab
root@pve.iespoveda.com's password:
pm153.keytab                                          100%  358   533.5KB/s   00:00
```

Ahora e el cliente:

```
chown root:root /etc/krb5.keytab
chmod 600 /etc/krb5.keytab
apt update
apt install libpam-krb5 libnss-ldap libpam-ldap nslcd
```

Verificamos que seguimos teniendo:

```
cat /etc/krb5.conf
[libdefaults]
    default_realm = IESPOVEDA.COM
    dns_lookup_realm = false
    dns_lookup_kdc = true

[realms]
    IESPOVEDA.COM = {
        kdc = ipa.iespoveda.com
        admin_server = ipa.iespoveda.com
    }

[domain_realm]
    .iespoveda.com = IESPOVEDA.COM
    iespoveda.com = IESPOVEDA.COM
```

Verificamos también:

```
cat /etc/nslcd.conf
uid nslcd
gid nslcd
uri ldap://ipa.iespoveda.com/
base dc=iespoveda,dc=com
tls_cacertfile /etc/ssl/certs/ca-certificates.crt
```

```
systemctl status nslcd
systemctl restart nslcd
kinit admin
```

```
root@pm153:~# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: admin@IESPOVEDA.COM

Valid starting       Expires              Service principal
03/26/2025 15:39:15  03/27/2025 15:16:30  krbtgt/IESPOVEDA.COM@IESPOVEDA.COM
```

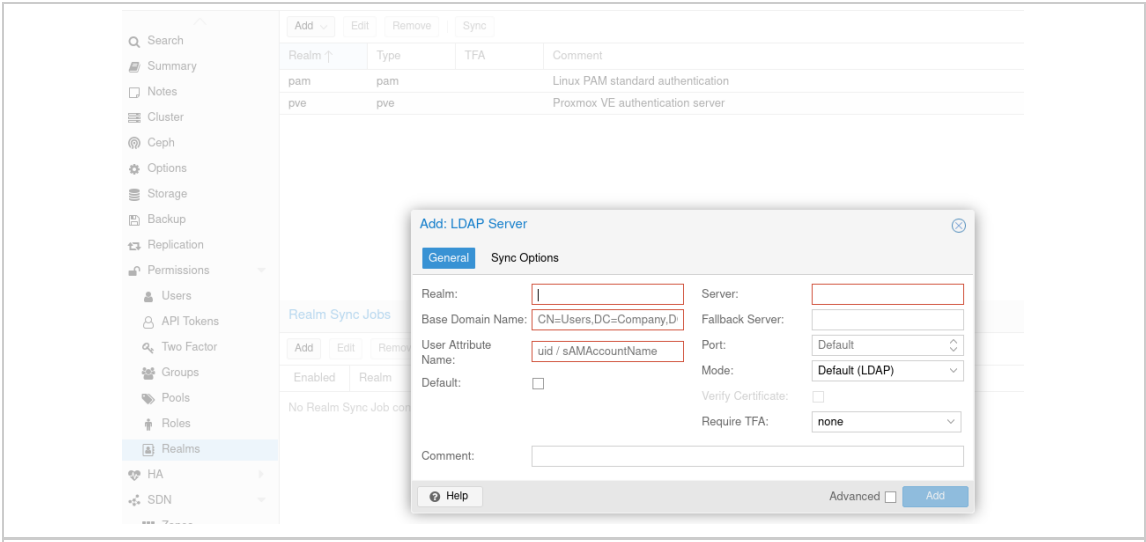
Verifica ipa server los usuario

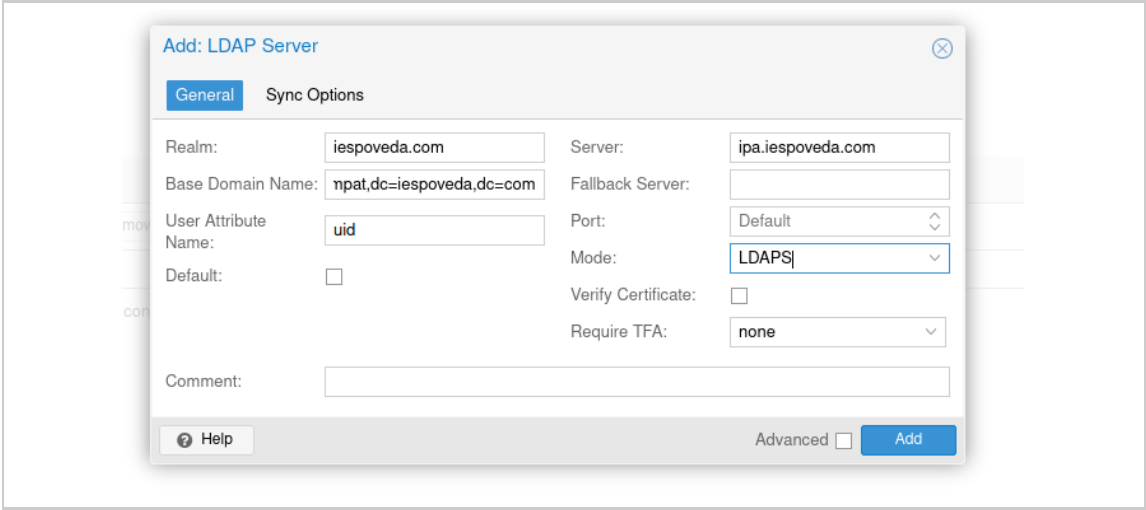
```
[root@ipa ~]# kinit admin
Password for admin@IESPOVEDA.COM:
[root@ipa ~]# systemctl status ipa
● ipa.service - Identity, Policy, Audit
     Loaded: loaded (/usr/lib/systemd/system/ipa.service; enabled; preset: disa>
     Active: active (exited) since Thu 2025-03-27 11:17:18 CET; 1h 0min ago
...
...
[root@ipa ~]# systemctl status krb5kdc.service
● krb5kdc.service - Kerberos 5 KDC
     Loaded: loaded (/usr/lib/systemd/system/krb5kdc.service; disabled; preset:>
     Active: active (running) since Thu 2025-03-27 11:15:46 CET; 1h 2min ago
    Process: 1821 ExecStart=/usr/sbin/krb5kdc -P /run/krb5kdc.pid $KRB5KDC_ARGS>
   Main PID: 1822 (krb5kdc)
 ...
 ...
[root@ipa ~]# kadmin.local
Authenticating as principal admin/admin@IESPOVEDA.COM with password.
kadmin.local:  listprincs
admin@IESPOVEDA.COM
K/M@IESPOVEDA.COM
krbtgt/IESPOVEDA.COM@IESPOVEDA.COM
kadmin/admin@IESPOVEDA.COM
kadmin/changepw@IESPOVEDA.COM
ldap/ipa.iespoveda.com@IESPOVEDA.COM
host/ipa.iespoveda.com@IESPOVEDA.COM
WELLKNOWN/ANONYMOUS@IESPOVEDA.COM
dogtag/ipa.iespoveda.com@IESPOVEDA.COM
HTTP/ipa.iespoveda.com@IESPOVEDA.COM
DNS/ipa.iespoveda.com@IESPOVEDA.COM
ipa-dnskeysyncd/ipa.iespoveda.com@IESPOVEDA.COM
host/pm153.iespoveda.com@IESPOVEDA.COM
arcadio@IESPOVEDA.COM
```

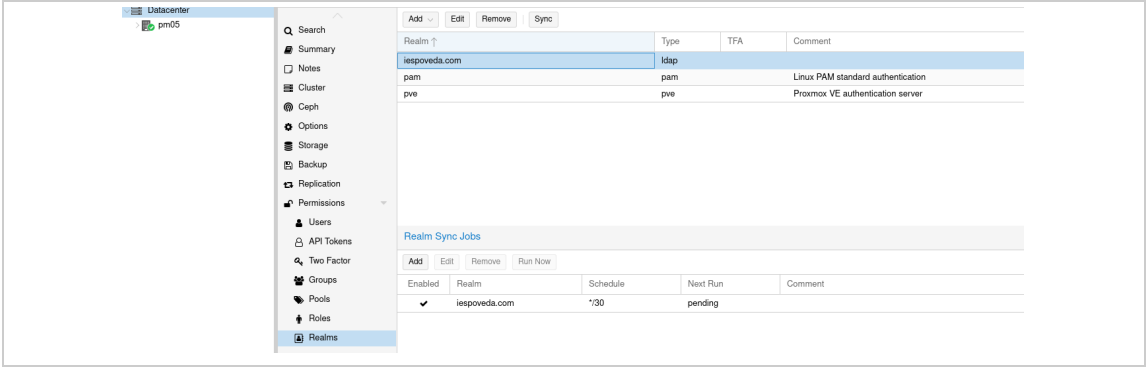
Realm --- Sincronizar usuarios con IPA server

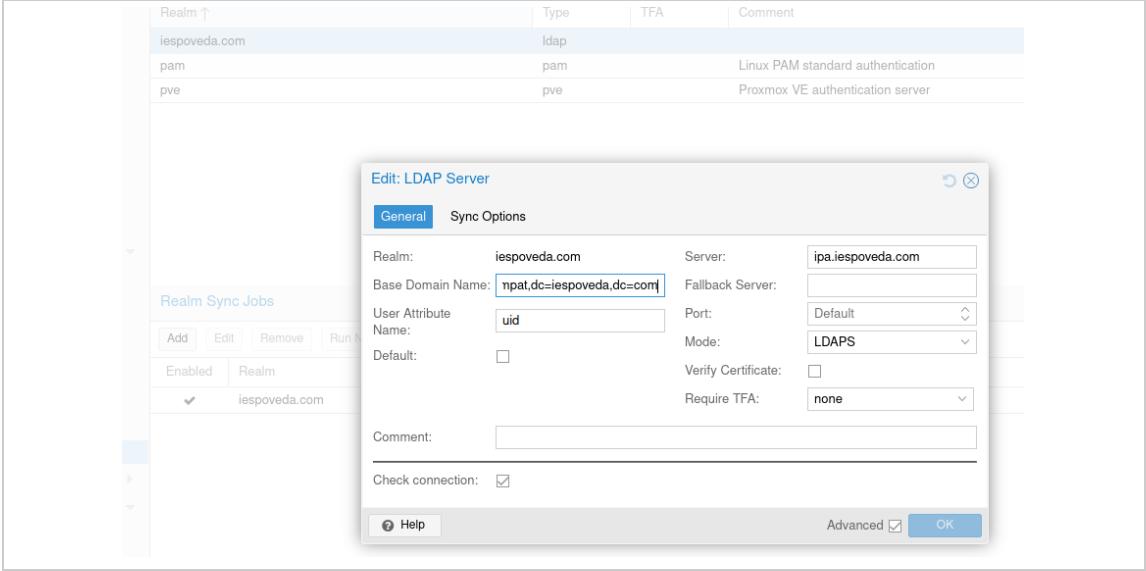
Para ver cómo hacerlo

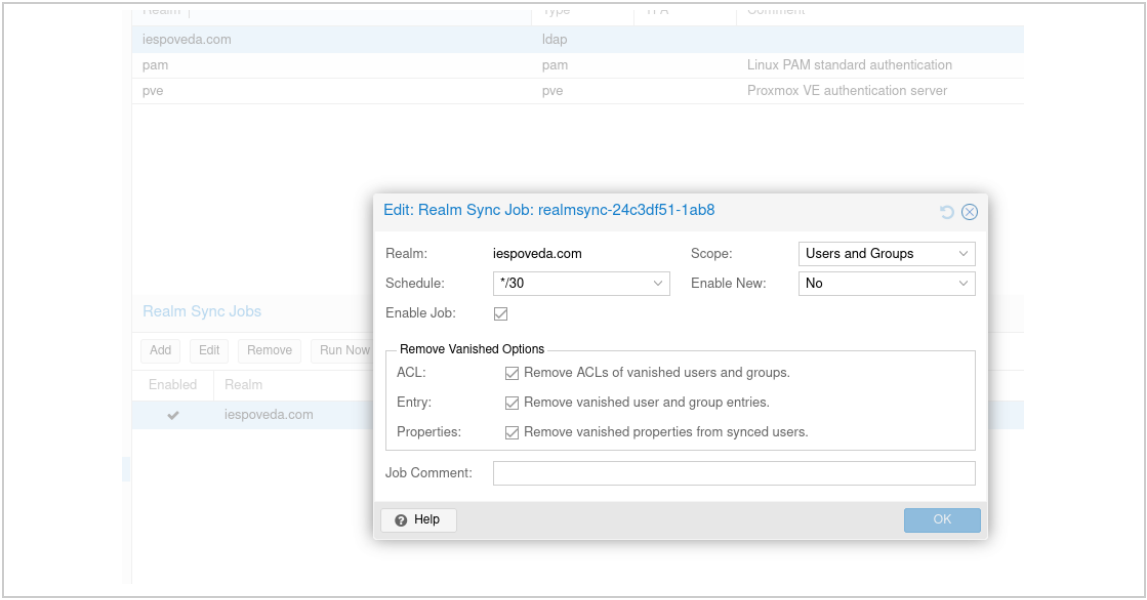
```
root@pm153:~# ldapsearch -x -H ldap://ipa.iespoveda.com -b
"cn=users,cn=compat,dc=iespoveda,dc=com"
c=iespoveda,dc=com"
# extended LDIF
#
# LDAPv3
# base <cn=users,cn=compat,dc=iespoveda,dc=com> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# users, compat, iespoveda.com
dn: cn=users,cn=compat,dc=iespoveda,dc=com
objectClass: extensibleObject
cn: users

# arcadio, users, compat, iespoveda.com
dn: uid=arcadio,cn=users,cn=compat,dc=iespoveda,dc=com
objectClass: posixAccount
objectClass: ipaOverrideTarget
objectClass: top
gecos: Arcadio Ortega
cn: Arcadio Ortega
uidNumber: 514000003
gidNumber: 514000003
loginShell: /bin/sh
homeDirectory: /home/arcadio
ipaAnchorUUID:: OklQQTppZXNwb3ZlZGEuY29tOjQ3YjhlNTZlLTBhZjUtMTFmMC04ZDQ1LWQ4Ym
 JjMWRjZWE0ZQ==
uid: arcadio

# admin, users, compat, iespoveda.com
dn: uid=admin,cn=users,cn=compat,dc=iespoveda,dc=com
objectClass: posixAccount
objectClass: ipaOverrideTarget
objectClass: top
gecos: Administrator
cn: Administrator
uidNumber: 514000000
gidNumber: 514000000
loginShell: /bin/bash
homeDirectory: /home/admin
ipaAnchorUUID:: OklQQTppZXNwb3ZlZGEuY29tOjkzYTgxNTI4LTBhNDQtMTFmMC1hMzFmLWQ4Ym
 JjMWRjZWE0ZQ==
uid: admin

# search result
search: 2
result: 0 Success

# numResponses: 4
# numEntries: 3

</cn=users,cn=compat,dc=iespoveda,dc=com>
```


Add: LDAP Server
General
Sync Options
Realm:
Base Domain Name:
CN=Users,DC=Company,D
User Attribute Name:
uid / sAMAccountName
Default:
Server:
Fallback Server:
Port:
Default
Mode:
Default (LDAP)
Verify Certificate:
Require TFA:
none
Comment:
Help
Advanced
Add

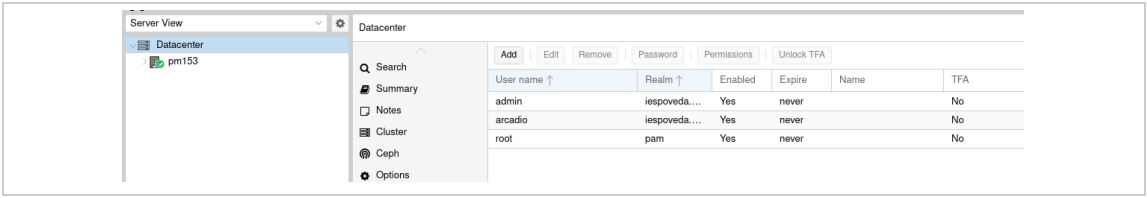

Add: LDAP Server
General
Sync Options
Realm:
iespoveda.com
Base Domain Name:
npat,dc=iespoveda,dc=com
User Attribute Name:
uid
Default:
Server:
ipa.iespoveda.com
Fallback Server:
Port:
Default
Mode:
LDAPS
Verify Certificate:
Require TFA:
none
Comment:
Help
Advanced
Add


Datacenter
pm05
Search
Summary
Notes
Cluster
Ceph
Options
Storage
Backup
Replication
Permissions
Users
API Tokens
Two Factor
Groups
Pools
Roles
Realms
Add
Edit
Remove
Sync
Realm
Type
TFA
Comment
iespoveda.com
ldap
pam
pam
Linux PAM standard authentication
pve
pve
Proxmox VE authentication server
Realm Sync Jobs
Add
Edit
Remove
Run Now
Enabled
Realm
Schedule
Next Run
Comment
iespoveda.com
*/30
pending


Realm
Type
TFA
Comment
iespoveda.com
ldap
pam
pam
Linux PAM standard authentication
pve
pve
Proxmox VE authentication server
Realm Sync Jobs
Add
Edit
Remove
Run Now
Enabled
Realm
iespoveda.com
Edit: LDAP Server
General
Sync Options
Realm:
iespoveda.com
Base Domain Name:
npat,dc=iespoveda,dc=com
User Attribute Name:
uid
Default:
Server:
ipa.iespoveda.com
Fallback Server:
Port:
Default
Mode:
LDAPS
Verify Certificate:
Require TFA:
none
Comment:
Check connection:
Help
Advanced
OK

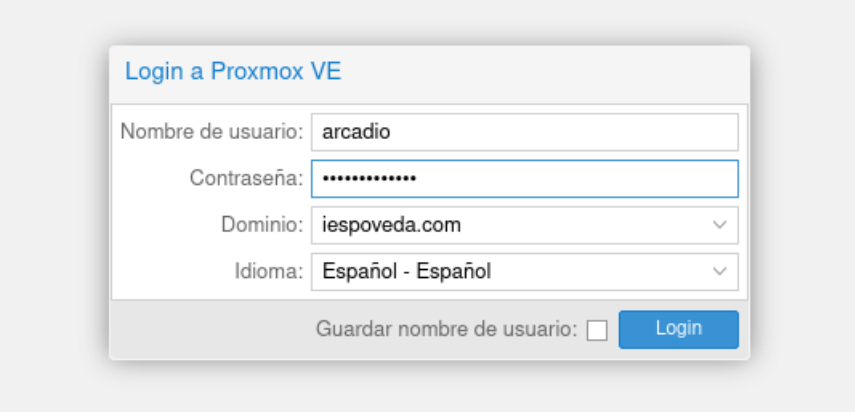


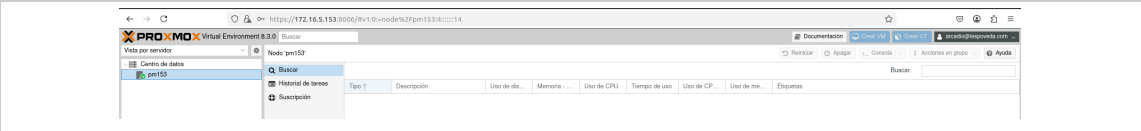
Después de actualizar tenemos:



El resultado de la gestón en la configuración debe dejar:

```
root@pm153:~# cat /etc/pve/domains.cfg
pve: pve
 comment Proxmox VE authentication server

pam: pam
 comment Linux PAM standard authentication

ldap: iespoveda.com
 base_dn cn=users,cn=compat,dc=iespoveda,dc=com
 server1 ipa.iespoveda.com
 user_attr uid
 default 0
 mode ldaps
```





Enlaces de interés https://forum.proxmox.com/threads/configuring-freeipa-ldap-server-with-proxmox.113500/

Nodos

Cliente sólo browser (firefox modo kiosk)

```
apt update
apt upgrade
apt install --no-install-recommends xorg openbox lightdm
```

```
apt install firefox-esr
```

PREVIAMENTE VEMOS SI TENEMOS AÚN TICKET

```
root@pm153:~# klist
klist: No credentials cache found (filename: /tmp/krb5cc_0)
root@pm153:~# kinit admin
Password for admin@IESPOVEDA.COM:
root@pm153:~# klist
Ticket cache: FILE:/tmp/krb5cc_0
Default principal: admin@IESPOVEDA.COM

Valid starting       Expires              Service principal
03/27/2025 12:08:26  03/28/2025 11:11:15  krbtgt/IESPOVEDA.COM@IESPOVEDA.COM
```

IMPORTANTE: AL AÑADIR EL USUARIO NO DAMOS CLAVE KERBEROS PARA QUE NO LO TENGA EN CUENTA,SÍ LA CLAVE EN LOCAL

```
adduser usuario
```

"Current Kerberos password" al ejecutar adduser en un sistema configurado con Kerberos, se está refiriendo a la contraseña de Kerberos que corresponde al usuario administrador o a un usuario con privilegios de administración en el sistema (por lo general, un usuario que ya esté autenticado en el sistema Kerberos).
IMPORTANTE: No le asignamos shell al usuario
`usuario:x:1000:1000:,,,:/home/usuario:/usr/bin/nologin`

```
vi /etc/lightdm/lightdm.conf
[LightDM]
[Seat:*]
#autologin-user=usuario
autologin-session=openbox
[XDMCPServer]
[VNCServer]
```

```
mkdir -p /home/usuario/.config/openbox
```

```
vi /home/usuario/.config/openbox/autostart
firefox-esr --kiosk http://localhost:8006
```

```
chown -R usuario:usuario /home/usuario/.config
```

Añadir un segundo proxmox a ipa server

En el server

Instalamos **pm154** con ip address 172.16.5.154

```
ipa dnsrecord-add iespoveda.com pm154 --a-ip-address=172.16.5.154
ipa host-add pm154.iespoveda.com
ipa-getkeytab -p host/pm154.iespoveda.com -k /etc/krb5.keytab
```

En el cliente

```
apt update
mkdir -p /etc/ipa
chown root:root /etc/ipa
chmod 755 /etc/ipa
wget -O /etc/ipa/ca.crt http://ipa.iespoveda.com/ipa/config/ca.crt
touch /etc/krb5.conf
```

```
vi /etc/krb5.conf
[libdefaults]
    default_realm = IESPOVEDA.COM
    dns_lookup_realm = false
    dns_lookup_kdc = true

[realms]
    IESPOVEDA.COM = {
        kdc = ipa.iespoveda.com
        admin_server = ipa.iespoveda.com
    }

[domain_realm]
    .iespoveda.com = IESPOVEDA.COM
    iespoveda.com = IESPOVEDA.COM
```

```
apt install freeipa-client
touch /etc/ipa/default.conf
```

```
vi /etc/ipa/default.conf
[global]
  domain = iespoveda.com
  hostname = pm154.iespoveda.com
  realm = IESPOVEDA.COM
  ipaclient = True
  server = ipa.iespoveda.com
  ca_cert = /etc/ipa/ca.crt
```

```
apt-get install krb5-kdc
kdb5_util create -s
systemctl start krb5-kdc
echo | openssl s_client -showcerts -connect ipa.iespoveda.com:443 2>/dev/null | openssl x509
-outform PEM > ipa_cert.pem
cp ipa_cert.pem /usr/local/share/ca-certificates/ipa_cert.crt
update-ca-certificates
scp /tmp/pm154.keytab root@ipa.iespoveda.com:/etc/krb5.keytab
```

En el servidor

```
ipa-getkeytab -p host/pm154.iespoveda.com -k /tmp/pm154.keytab
scp /tmp/pm154.keytab root@pm154.iespoveda.com:/etc/krb5.keytab
```

En el cliente

```
chown root:root /etc/krb5.keytab
chmod 600 /etc/krb5.keytab
install libpam-krb5 libnss-ldap libpam-ldap nslcd
systemctl restart nslcd
kinit admin
klist
```

Cliente sólo browser (firefox modo kiosk)

```
apt update
apt upgrade
apt install --no-install-recommends xorg openbox lightdm
apt install firefox-esr
adduser usuario
```

```
vi /etc/lightdm/lightdm.conf
[LightDM]
[Seat:*]
#autologin-user=usuario
autologin-session=openbox
[XDMCPServer]
[VNCServer]
```

```
echo "firefox-esr --kiosk http://localhost:8006" > /home/usuario/.config/openbox/autostart
chown -R usuario:usuario /home/usuario/.config
```

Unir un Ubuntu Server a un IPA Server (FreeIPA) como cliente

Pre-requisitos

- El Ubuntu Server debe tener conectividad de red con el IPA Server.
- Configura correctamente la resolución DNS (el IPA Server normalmente también funciona como servidor DNS).
- Sincroniza el tiempo con el IPA Server (usa chrony o ntp).
- Asegúrate de tener permisos de administrador en el IPA Server.

Instala los paquetes necesarios en Ubuntu

```
sudo apt update
sudo apt install freeipa-client sssd realmd oddjob oddjob-mkhomedir adcli samba-common krb5-
user
```

Durante la instalación se te pedirá el dominio Kerberos y el servidor KDC. Puedes poner los del IPA Server (por ejemplo, IPA.DOMINIO.LOCAL y ipa.dominio.local).

Unir el Ubuntu Server al IPA Server

Ejecuta:

```
sudo ipa-client-install --hostname=$(hostname -f) --server=ipa.iespoveda.com --
domain=iespoveda.com --realm=IESPOVEDA.COM
```

Verifica la unión al dominio

```
id usuario@IESPOVEDA.COM
```

Si devuelve los datos del usuario, todo está bien.

Configura el acceso para los usuarios del IPA

Edita /etc/sssd/sssd.conf si es necesario, pero normalmente ya está bien configurado tras la instalación. Asegúrate de que pam_mkhomedir.so esté habilitado en PAM:

```
sudo pam-auth-update
```

Marca la opción “Create home directory on login”.

Prueba iniciar sesión con un usuario del IPA

Puedes probar con SSH:

```
ssh usuario@ubuntu-server
```

TrueNas IPA

En IPA sever

```
[root@ipa ~]# ipa dnsrecord-add iespoveda.com truenas --a-ip-address=172.16.5.111
  Record name: truenas
  A record: 172.16.5.111
```

```
[root@ipa ~]# ipa host-add truenas.iespoveda.com
Added host "truenas.iespoveda.com"

  Host name: truenas.iespoveda.com
  Principal name: host/truenas.iespoveda.com@IESPOVEDA.COM
  Principal alias: host/truenas.iespoveda.com@IESPOVEDA.COM
  Password: False
  Keytab: False
  Managed by: truenas.iespoveda.com
```

```
[root@ipa ~]# ipa-getkeytab -p host/truenas.iespoveda.com -k /etc/krb5.keytab
```

Tiene que quedar

```
cat /etc/krb5.conf
#
# krb5.conf(5) - configuration file for Kerberos 5
# $FreeBSD$
#

[appdefaults]
            pam = {
                   forwardable = true
                   ticket_lifetime = 86400
                   renew_lifetime = 86400
            }

[libdefaults]
            dns_lookup_realm = true
            dns_lookup_kdc = true
            ticket_lifetime = 24h
            clockskew = 300
            forwardable = true

[domain_realm]
            ipa.iespoveda.com = ipa.iespoveda.com
            .ipa.iespoveda.com = ipa.iespoveda.com
            IPA.IESPOVEDA.COM = ipa.iespoveda.com
            .IPA.IESPOVEDA.COM = ipa.iespoveda.com

[realms]
            ipa.iespoveda.com = {
                   default_domain = ipa.iespoveda.com
                   kdc = ipa.iespoveda.com
                   admin_server = root
                   kpasswd_server = xxxxxxxxxxxxxxxxxxxxxxxxxxxx
            }

[logging]
            default = SYSLOG:INFO:LOCAL7
```

Probamos:

```
getent passwd
...
...
arcadio:*:514000003:514000003:Arcadio Ortega:/home/arcadio:/bin/sh
admin:*:514000000:514000000:Administrator:/home/admin:/bin/sh
admin:*:514000000:514000000:Administrator:/home/admin:/bin/sh
arcadio:*:514000003:514000003:Arcadio Ortega:/home/arcadio:/bin/sh
```

o

```
ldapsearch -x -LLL -H ldap://ipa.iespoveda.com -b "cn=users,cn=accounts,dc=iespoveda,dc=com"

dn: cn=users,cn=accounts,dc=iespoveda,dc=com
objectClass: top
objectClass: nsContainer
cn: users

dn: uid=admin,cn=users,cn=accounts,dc=iespoveda,dc=com
objectClass: top
objectClass: person
objectClass: posixaccount
objectClass: krbprincipalaux
objectClass: krbticketpolicyaux
objectClass: inetuser
objectClass: ipaobject
objectClass: ipasshuser
objectClass: ipaSshGroupOfPubKeys
objectClass: ipaNTUserAttrs
uid: admin
cn: Administrator
sn: Administrator
uidNumber: 514000000
gidNumber: 514000000
homeDirectory: /home/admin
loginShell: /bin/bash
gecos: Administrator
ipaNTSecurityIdentifier: S-1-5-21-483169404-2447074518-752674619-500

dn: uid=arcadio,cn=users,cn=accounts,dc=iespoveda,dc=com
givenName: Arcadio
sn: Ortega
uid: arcadio
cn: Arcadio Ortega
displayName: Arcadio Ortega
initials: AO
gecos: Arcadio Ortega
objectClass: top
objectClass: person
objectClass: organizationalperson
objectClass: inetorgperson
objectClass: inetuser
objectClass: posixaccount
objectClass: krbprincipalaux
objectClass: krbticketpolicyaux
objectClass: ipaobject
objectClass: ipasshuser
objectClass: ipaSshGroupOfPubKeys
objectClass: mepOriginEntry
objectClass: ipantuserattrs
loginShell: /bin/sh
homeDirectory: /home/arcadio
uidNumber: 514000003
gidNumber: 514000003
ipaNTSecurityIdentifier: S-1-5-21-483169404-2447074518-752674619-1003
```

Dar permiso sudo a un usuario en FreeIPA

Una vez creado el usuario "arcadio" le damos permiso de sudo, a través del grupo que creamos sysadmin

```
[root@ipa ~]# ipa user-find arcadio
--------------
1 user matched
--------------
  User login: arcadio
  First name: Arcadio
  Last name: Ortega Reinoso
  Home directory: /home/arcadio
  Login shell: /bin/sh
  Principal name: arcadio@IESPOVEDA.COM
  Principal alias: arcadio@IESPOVEDA.COM
  Email address: arcadio@iespoveda.com
  UID: 1772000003
  GID: 1772000003
  Account disabled: False
----------------------------
Number of entries returned 1
----------------------------
[root@ipa ~]#  authselect enable-feature with-sudo
Make sure that SSSD service is configured and enabled. See SSSD documentation for more
information.

[root@ipa ~]# systemctl restart sssd
[root@ipa ~]# authselect current
Profile ID: sssd
Enabled features:
- with-sudo
[root@ipa ~]# ipa group-add --desc='Sysadmin Team' sysadmin
----------------------
Added group "sysadmin"
----------------------
  Group name: sysadmin
  Description: Sysadmin Team
  GID: 1772000004
[root@ipa ~]# ipa group-find sysadmin
---------------
1 group matched
---------------
  Group name: sysadmin
  Description: Sysadmin Team
  GID: 1772000004
----------------------------
Number of entries returned 1
----------------------------
[root@ipa ~]# ipa sudorule-add sysadmin_sudo --hostcat=all --runasusercat=all --
runasgroupcat=all --cmdcat=all
-------------------------------
Added Sudo Rule "sysadmin_sudo"
-------------------------------
  Rule name: sysadmin_sudo
  Enabled: True
  Host category: all
  Command category: all
  RunAs User category: all
  RunAs Group category: all
[root@ipa ~]# ipa sudorule-add-user sysadmin_sudo --group sysadmin
  Rule name: sysadmin_sudo
```

```
  Rule name: sysadmin_sudo
  Enabled: True
  Host category: all
  Command category: all
  RunAs User category: all
  RunAs Group category: all
  User Groups: sysadmin
-------------------------
Number of members added 1
-------------------------
[root@ipa ~]# ipa sudorule-show sysadmin_sudo
  Rule name: sysadmin_sudo
  Enabled: True
  Host category: all
  Command category: all
  RunAs User category: all
  RunAs Group category: all
  User Groups: sysadmin
[root@ipa ~]# pa group-add-member --user=arcadio sysadmin
-bash: pa: command not found
[root@ipa ~]# ipa group-add-member --user=arcadio sysadmin
  Group name: sysadmin
  Description: Sysadmin Team
  GID: 1772000004
  Member users: arcadio
  Member of Sudo rule: sysadmin_sudo
-------------------------
Number of members added 1
-------------------------
[root@ipa ~]# ipa group-show sysadmin
  Group name: sysadmin
  Description: Sysadmin Team
  GID: 1772000004
  Member users: arcadio
  Member of Sudo rule: sysadmin_sudo
```

Si no tenemos sudo en el sistema entramos en modo recovery y lo instalamos

NFS server

```
apt install nfs-kernel-server
mkdir -p /mnt/nfs/{templates,backups}
```

Exportamos:

```
cat /etc/exports
# /etc/exports: the access control list for filesystems which may be exported
#  to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes       hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4        gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#
/mnt/nfs/templates 172.16.5.0/16(rw,sync,no_subtree_check)
/mnt/nfs/backups 171.16.0.0/16(rw,sync,no_subtree_check)
```

systemctl restart nfs-server exportfs Para realizar verificaciones instalamos:

```
apt install net-tools
```

Verificamos

```
ifconfig
netstat -nr
ip route
```

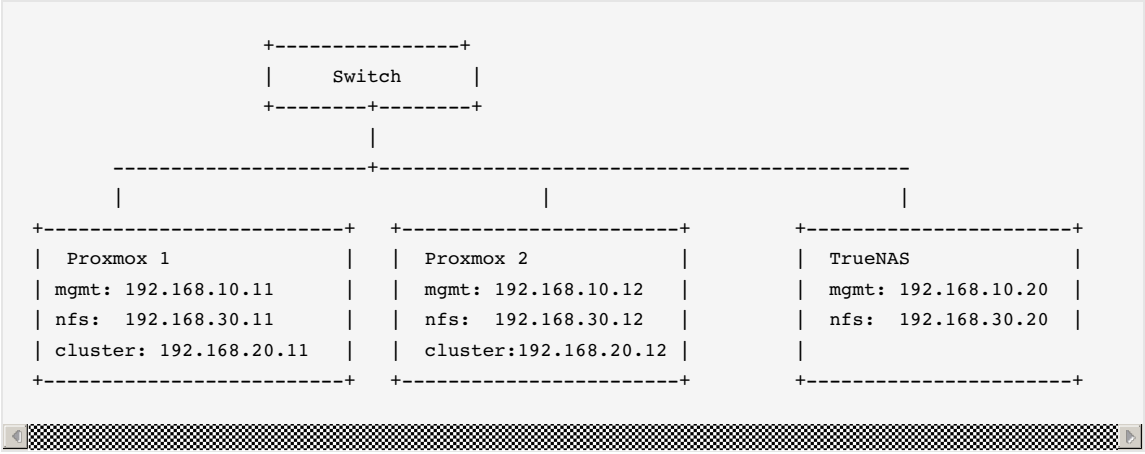
Propuesta sin VLAN para una clase

Cluster. Propuesta inicial

Cómo sería la arquitectura recomendada:

- Red de gestión para Proxmox (por ejemplo 192.168.10.0/24).
- Red de cluster (Corosync) dedicada (por ejemplo 192.168.20.0/24 o VLAN separada).
- Red de almacenamiento (NFS) dedicada (por ejemplo 192.168.30.0/24 o VLAN).

El servidor TrueNAS también debería tener una interfaz en la red NFS (o VLAN) para atender peticiones solo ahí.



Si podemos tener VLAN:

- 192.168.10.x: Red de gestión (Proxmox web UI, SSH, etc.)
- 192.168.20.x: Red del cluster (Corosync).
- 192.168.30.x: Red de almacenamiento (NFS entre Proxmox y TrueNAS).

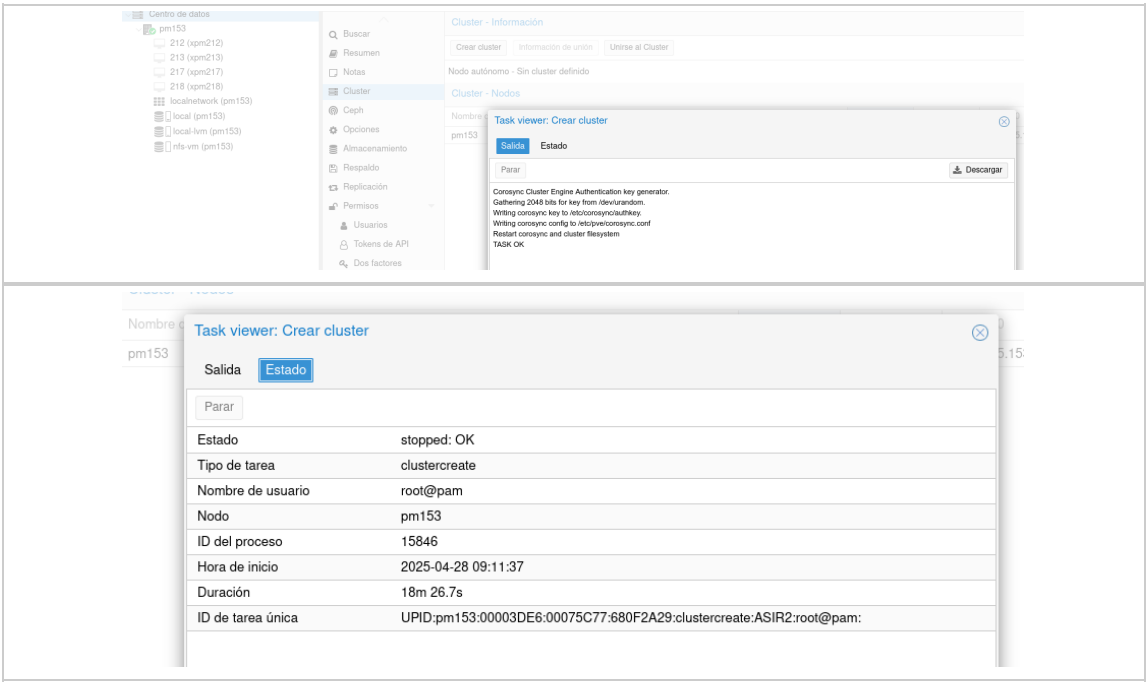
Si no podemos, como es nuestro caso, usamos el cableado para proxmox y la wifi para internet:

----->>> Y para la salida a internet vamos a usar la wifi

Cluster

Nodo master del cluster:

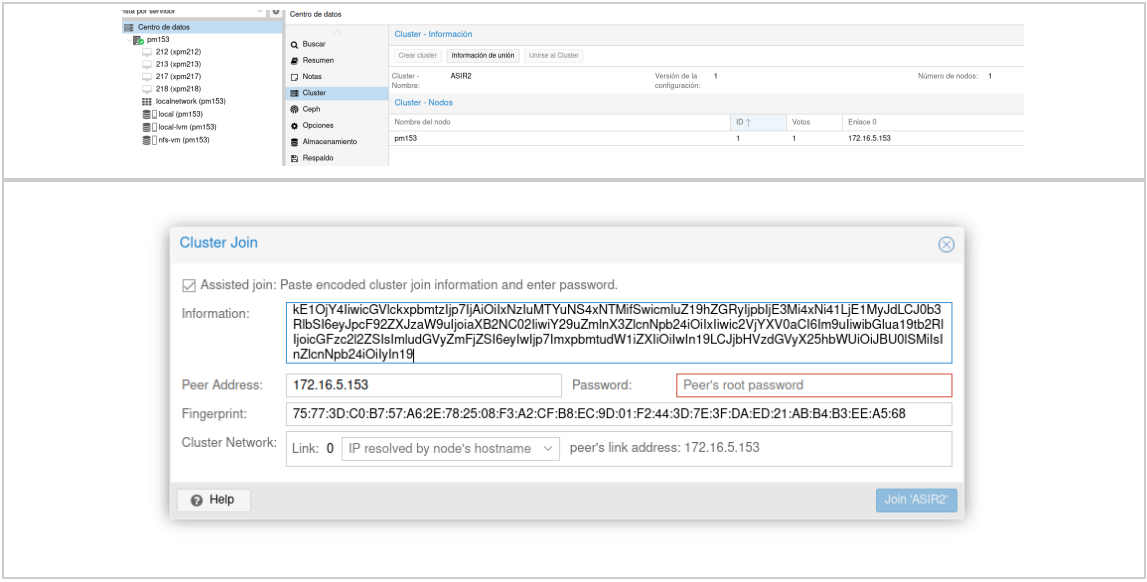
```
pvecm create NOMBREDELCLUSTER --bindnet0_addr 192.168.20.11
```

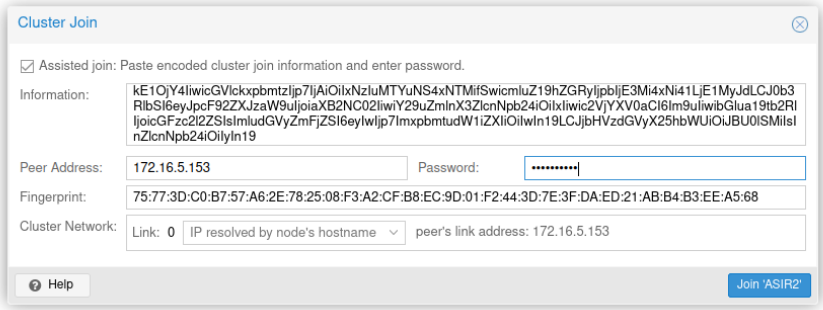


Nodos que se unen al cluster, en cada uno de los nodos ejecutamos:

```
pvecm add 192.168.20.11
```

El resultado lo tendremos en /etc/pve/corosync.conf

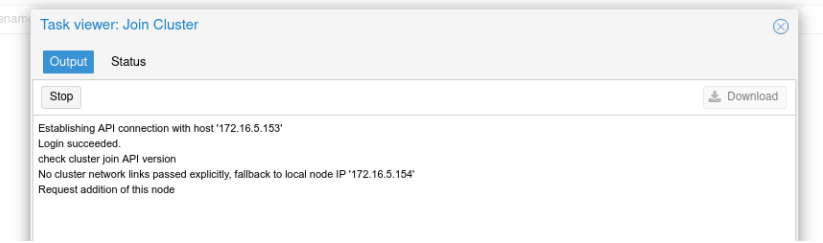




Si al trata de unir a un cluster proxmox obtenemos:

```
detected the following error(s):
* this host already contains virtual guests
TASK ERROR: Check if node may join a cluster failed!"
```

El error es porque el nodo que estás intentando agregar al clúster de Proxmox ya tiene máquinas virtuales (VMs) configuradas en él.
Esto impide que el nodo se una al clúster, ya que **Proxmox no permite que un nodo con VMs existentes se una a un clúster sin antes realizar una serie de pasos para asegurarse de que no haya conflictos**.




```
qm shutdown <ID_VM>
```

O si hay más de una VM:

```
qm shutdown <ID_VM1> <ID_VM2> <ID_VM3>
```

También podemos Mover las VMs a otro nodo:

```
qm migrate <ID_VM> <nombre_del_nodo_destino>
```

Destruir las VMs en el nodo:

```
qm destroy <ID_VM>
```

En nuestro caso no existe el filesystem dónde estaba

```
root@pm154:~# qm list
      VMID NAME                 STATUS     MEM(MB)    BOOTDISK(GB) PID
       100 xpm212               stopped    2048              40.00 0
root@pm154:~# qm shutdown 100
VM 100 not running

root@pm154:~# qm stop 100
root@pm154:~# qm destroy 100
storage 'lvm-scsi' does not exist
root@pm154:~# find / -name "100-*.qcow2"
root@pm154:~# find / -name "100-*.raw"
root@pm154:~# cat /etc/pve/storage.cfg
dir: local
        path /var/lib/vz
        content backup,iso,vztmpl

lvmthin: local-lvm
        thinpool data
        vgname pve
        content images,rootdir

nfs: nfs-vm
        export /mnt/truenasdisk/nfsvm
        path /mnt/pve/nfs-vm
        server 172.16.5.111
        content images
        nodes pm154
        prune-backups keep-all=1
```

```
root@pm154:~# cat /etc/pve/qemu-server/100.conf
agent: 1
boot: order=scsi0;ide2;net0
cores: 1
cpu: x86-64-v2-AES
ide2: local:iso/ubuntu-22.04.5-live-server-amd64.iso,media=cdrom,size=2086842K
memory: 2048
meta: creation-qemu=9.0.2,ctime=1744108130
name: xpm212
net0: virtio=BC:24:11:1E:09:7E,bridge=vmbr0,firewall=1
numa: 0
ostype: l26
scsi0: lvm-scsi:vm-100-disk-0,iothread=1,size=40G
scsihw: virtio-scsi-single
smbios1: uuid=109fcadb-9e80-4b12-b4c0-afc205300992
sockets: 1
vmgenid: 2f1de7bf-1c0b-4795-8c61-1f215722d225
```

Ekiminamos la línea:

```
 scsi0: lvm-scsi:vm-100-disk-0,iothread=1,size=40G
```

```
root@pm154:~# qm destroy 100
root@pm154:~# vgs
  VG  #PV #LV #SN Attr   VSize   VFree
  pve   1   3   0 wz--n- 237.47g 16.00g
root@pm154:~# lvs
  LV   VG  Attr       LSize    Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  data pve twi-aotz-- <141.23g             0.00   1.13
  root pve -wi-ao----  <69.37g
  swap pve -wi-ao----    8.00g
```

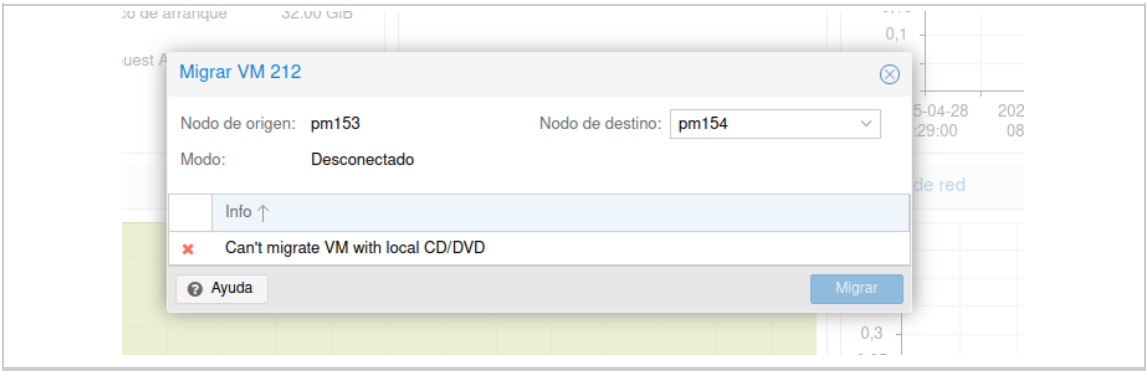
Ahora uniríamos el nodo al cluster y podemos verificarlo mediante:

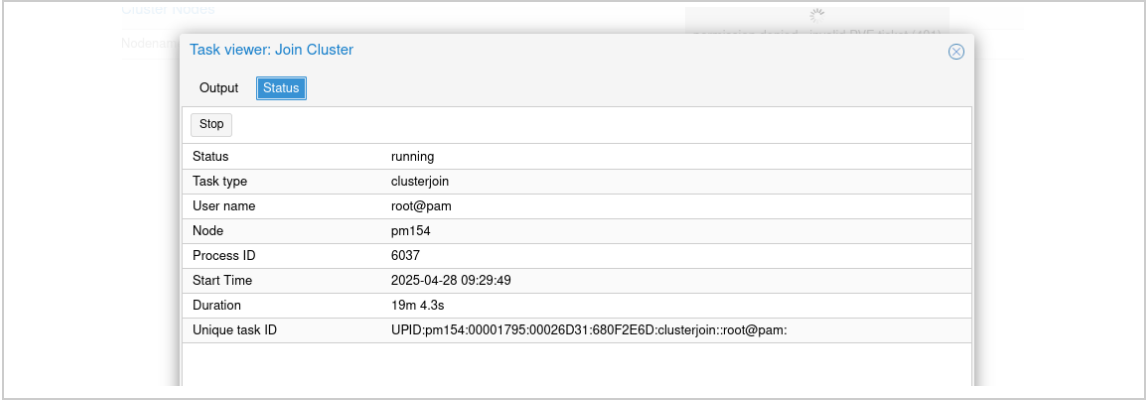
```
pvecm nodes
```

Ejemplo a la hora de migrar:

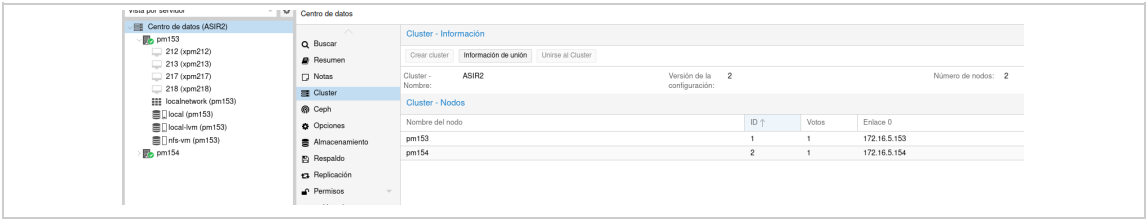
```
root@pm153:~# qm migrate 212  pm154
2025-04-28 09:39:55 starting migration of VM 212 to node 'pm154' (172.16.5.154)
2025-04-28 09:39:55 found local disk 'local:iso/ubuntu-24.10-live-server-amd64.iso'
(attached)
2025-04-28 09:39:55 can't migrate local disk 'local:iso/ubuntu-24.10-live-server-amd64.iso':
local cdrom image
2025-04-28 09:39:55 ERROR: Problem found while scanning volumes - can't migrate VM - check
log
2025-04-28 09:39:55 aborting phase 1 - cleanup resources
2025-04-28 09:39:55 ERROR: migration aborted (duration 00:00:00): Problem found while
scanning volumes - can't migrate VM - check log
migration aborted
```

Tengo que quitar primero el disco CD de la instalación:





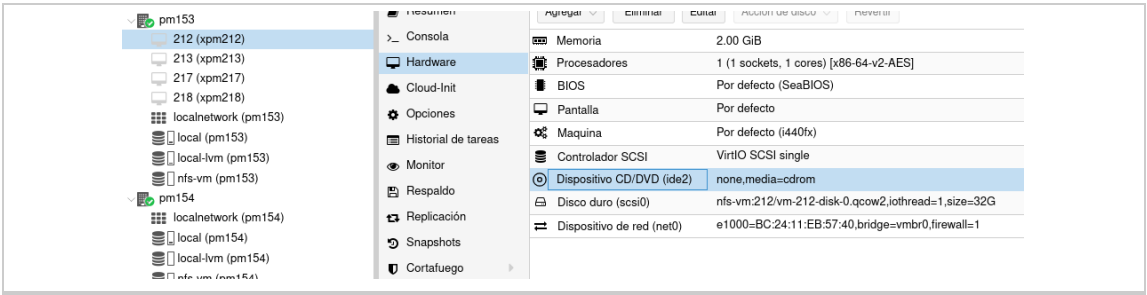
Verificamos nuestro clúster:

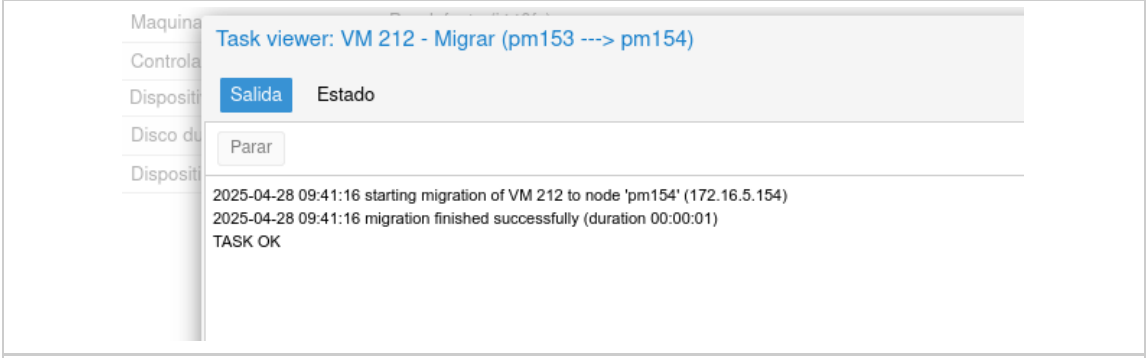


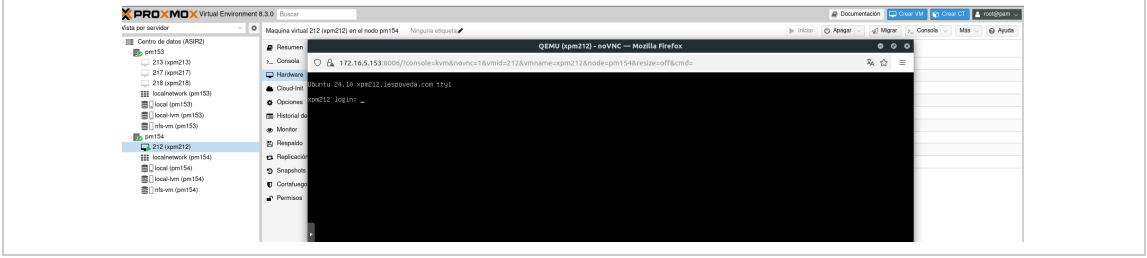
Ahora podemos migrar una áquina virtual entre elementos del clúster:

Lo primero es que tienen que tener acceso al mismo hardware y software

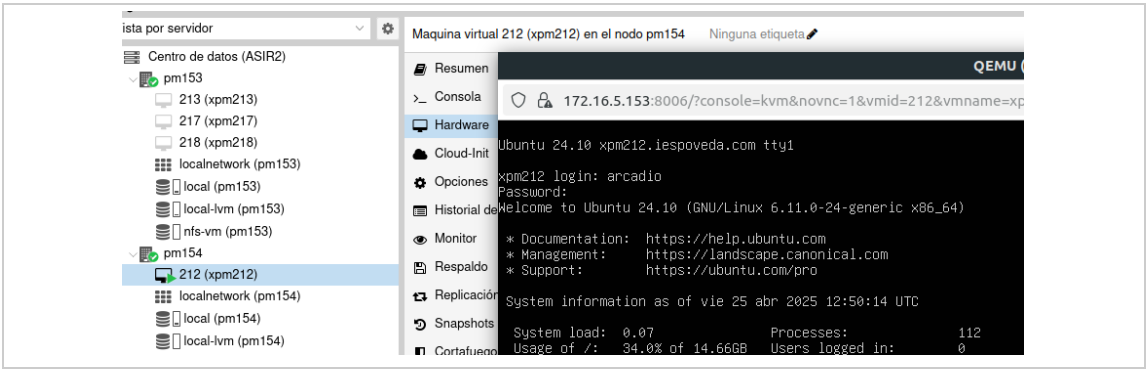
No tiene acceso el destino al iso de la distribución de ubuntu:







Podemos entrar con el usurio que tengamos dado de alta, en nuestro ejemplo dado de alta en ipa server y con home en nfs.



Ejemplo de NFS en truenas

```
cat /etc/pve/storage.cfg
nfs: truenas_nfs
    export /mnt/volumen_nfs
    path /mnt/pve/truenas_nfs
    server 192.168.30.20
    options vers=3
    content images,iso,vztmpl
```

En nuestro caso:

```
nfs: nfs-vm
        export /mnt/truenasdisk/nfsvm
        path /mnt/pve/nfs-vm
        server 172.16.5.111
        content images
        nodes pm153
        prune-backups keep-all=1
```

Gestión usuarios

Damos de alta el usuario en el ipa server

```
ipa user-add juan --first=Juan --last=Gomez --password
Password:
Enter Password again to verify:
-----------------
Added user "juan"
-----------------
  User login: juan
  First name: Juan
  Last name: Gomez
  Full name: Juan Gomez
  Display name: Juan Gomez
  Initials: JG
  Home directory: /home/juan
  GECOS: Juan Gomez
  Login shell: /bin/sh
  Principal name: juan@IESPOVEDA.COM
  Principal alias: juan@IESPOVEDA.COM
  User password expiration: 20250429120314Z
  Email address: juan@iespoveda.com
  UID: 514000004
  GID: 514000004
  Password: True
  Member of groups: ipausers
  Kerberos keys available: True
```

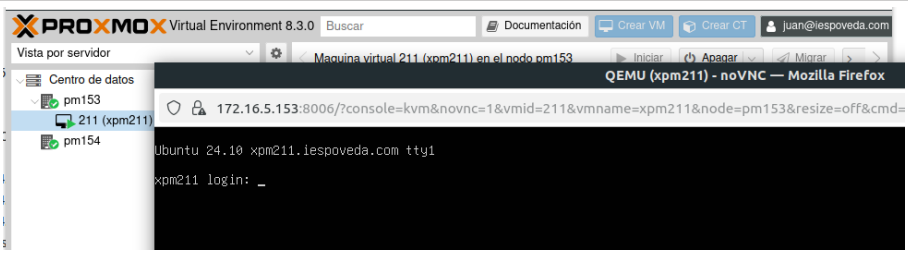
En proxmox:

```
# pveum user add juan@iespoveda.com
```

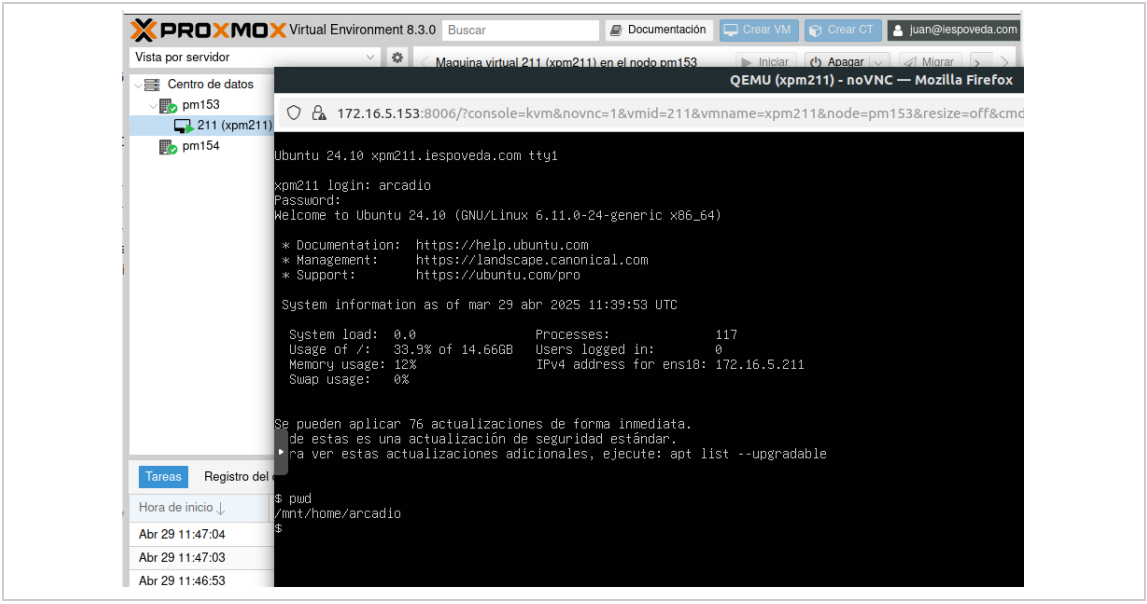
Le damos permiso de ejecución de una máquina en particular:

```
# pveum aclmod /vms/211 --user juan@iespoveda.com --role PVEVMUser
```

Podemos ver cómo entrando con el usuario creado puede ejecutar la máquina virtual para la que tiene permiso



Además lo ejecutará con el nombre de usuario que tenga cceso a la máquina virtual en cuestión.


PROXMOX Virtual Environment 8.3.0
Buscar
Documentación
Crear VM
Crear CT
juan@iespoveda.com
Vista por servidor
Maquina virtual 211 (xpm211) en el nodo pm153
Iniciar
Apagar
Migrar
Centro de datos
pm153
211 (xpm211)
pm154
QEMU (xpm211) - noVNC — Mozilla Firefox
172.16.5.153:8006/?console=kvm&novnc=1&vmid=211&vmname=xpm211&node=pm153&resize=off&cmd
Ubuntu 24.10 xpm211.iespoveda.com tty1
xpm211 login: arcadio
Password:
Welcome to Ubuntu 24.10 (GNU/Linux 6.11.0-24-generic x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro
System information as of mar 29 abr 2025 11:39:53 UTC
System load: 0.0
Usage of /: 33.9% of 14.66GB
Memory usage: 12%
Swap usage: 0%
Processes: 117
Users logged in: 0
IPv4 address for ens18: 172.16.5.211
Se pueden aplicar 76 actualizaciones de forma inmediata.
de estas es una actualización de seguridad estándar.
ra ver estas actualizaciones adicionales, ejecute: apt list --upgradable
$ pwd
/mnt/home/arcadio
$
Tareas
Registro del
Hora de inicio
Abr 29 11:47:04
Abr 29 11:47:03
Abr 29 11:46:53

Ver los roles

```
pveum role list
```

PVEVMUser

ver qué acciones permite:

```
pveum role list | grep PVEVMUser
```

Este rol permite:

- Encender, apagar, suspender y resetear VMs
- Ver el estado de las VMs
- No puede modificar discos, hardware, ni configuración

También podemos: Asignar el rol sobre un nodo completo:

```
pveum aclmod /nodes/pve-nodo2 --user juan@ipa --role PVEVMUser
```

PVEVMAdmin

```
pveum aclmod /nodes/pve-nodo2 --user juan@ipa --role PVEVMAdmin
```

Este rol permite:

- Crear, eliminar y administrar VMs
- No tiene acceso a Proxmox completo

Nodos del clúster

Eliminar Clusteres y/o sus nodos

Pasos para sacar un nodo de un cluster y eliminarlo posteriormente:

1. Inicia sesión en el nodo que deseas eliminar (usando SSH o la interfaz web de Proxmox).

2. Deshabilita las máquinas virtuales (si es necesario). Puedes migrar las máquinas virtuales a otro nodo si es necesario, usando la interfaz web de Proxmox o desde la línea de comandos:

```
qm migrate VMID <destination-node> --online
```

Reemplaza `VMID` con el ID de la máquina virtual y `<destination-node>` con el nombre del nodo al que quieres migrarla.

3. Elimina el nodo del cluster usando el siguiente comando:

```
pvecm delnode <nombre-del-nodo>
```

Reemplaza `<nombre-del-nodo>` con el nombre del nodo que deseas eliminar del cluster. Este comando eliminará al nodo del cluster, y se actualizarán las configuraciones de quorum y corosync automáticamente.

4. Verifica que el nodo se haya eliminado del cluster:

Después de ejecutar el comando, puedes verificar que el nodo ya no está en el cluster con:

```
pvecm status
```

Esto te mostrará los nodos restantes del cluster.

Eliminar la configuración del cluster en el nodo eliminado

Ahora que el nodo ya no es parte del cluster, debes limpiar la configuración de cluster (Corosync) de ese nodo.

1. Detén los servicios de cluster en el nodo que quieres eliminar:

```
systemctl stop pve-cluster
systemctl stop corosync
```

2. Elimina la configuración de cluster:

Para borrar la configuración de Proxmox y Corosync, debes eliminar las configuraciones de cluster:

```
rm -rf /etc/pve/*
rm -rf /etc/corosync/*
```

3. Reinicia el nodo:

Después de borrar la configuración, reinicia el nodo para asegurarte de que se haya limpiado todo correctamente.

```
reboot
```

Dar de baja un clúster

Para sacar un nodo de un cluster de Proxmox y luego eliminar el cluster completamente, necesitas seguir una serie de pasos.

Eliminar el cluster en los nodos restantes

Si deseas eliminar todo el cluster (por ejemplo, si quieres destruir completamente el clúster en todos los nodos), repite el proceso de eliminación del cluster en todos los nodos restantes, usando los mismos comandos en cada nodo.

```
systemctl stop pve-cluster
systemctl stop corosync
rm -rf /etc/pve/*
rm -rf /etc/corosync/*
reboot
```

Cómo añadir un nuevo nodo a un clúster Proxmox con servidor NTP personalizado

1. **Instalar Proxmox VE**

Instala Proxmox VE normalmente desde la ISO oficial o sobre una instalación base de Debian.

2. **Configurar NTP con el servidor interno (ipa.iespoveda.com)**

```
nano /etc/chrony/chrony.conf
```

Comenta el pool público y añade tu servidor NTP:

```
# pool 2.debian.pool.ntp.org iburst
server ipa.iespoveda.com iburst
```

Reinicia Chrony:

```
systemctl restart chrony
```

Verifica que el servidor esté activo:

```
chronyc sources -v
```

Busca una línea con `^*` al inicio, indicando que el servidor ipa.iespoveda.com está siendo usado.

3. **Forzar sincronización y habilitar integración con systemd**

```
systemctl enable chrony-wait
systemctl restart chrony
chronyc -a makestep
timedatectl
```

Si `System clock synchronized: no`, ejecuta:

```
timedatectl set-ntp false
timedatectl set-ntp true
systemctl daemon-reexec
systemctl daemon-reload
systemctl restart chrony
chronyc -a makestep
timedatectl
```

Deberías ver ahora: `System clock synchronized: yes`.

4. **Añadir el nodo al clúster**

Desde un nodo del clúster, ejecuta:

```
pvecm add 172.16.5.154
```

Sustituye la IP por la del nuevo nodo. Acepta el fingerprint y sigue las instrucciones para completar la unión.

En un sólo fichero:

```
systemctl restart chrony
chronyc sources -v
systemctl enable chrony-wait
systemctl restart chrony
chronyc -a makestep
timedatectl
timedatectl set-ntp false
timedatectl set-ntp true
systemctl daemon-reexec
systemctl daemon-reload
systemctl restart chrony
chronyc -a makestep
timedatectl
```

Podemos probar a ejecutar el siguiente shell script (planteado por ChatGPT):

```
#!/bin/bash

echo "[+] Desactivando systemd-timesyncd..."
systemctl stop systemd-timesyncd 2>/dev/null || true
systemctl disable systemd-timesyncd 2>/dev/null || true

echo "[+] Activando chronyd y el servicio chrony-wait..."
systemctl enable --now chronyd
systemctl enable chrony-wait

# Pausa para permitir que chronyd arranque y adquiera servidores
sleep 3

echo "[+] Ejecutando primer makestep..."
chronyc -a makestep

# Espera extra para asegurar que chronyd establezca la hora correctamente
sleep 2

echo "[+] Verificando fuentes de tiempo..."
chronyc sources -v

echo "[+] Configurando timedatectl para usar NTP..."
timedatectl set-ntp false
timedatectl set-ntp true

# Reinicio de daemon para asegurarnos de que systemd toma los cambios
echo "[+] Reiniciando daemons de systemd..."
systemctl daemon-reexec
systemctl daemon-reload

# Pausa corta tras reinicio de daemons
sleep 2

echo "[+] Reiniciando chronyd por última vez..."
systemctl restart chronyd

# Segundo makestep para garantizar sincronización
echo "[+] Ejecutando segundo makestep (final)..."
chronyc -a makestep

# Confirmar estado de sincronización
echo "[+] Estado final de la hora del sistema:"
timedatectl

echo "[✔] Sincronización NTP finalizada correctamente."
```

AULA

Servidor previo

EN NUESTRO IPA SERVER HACEMOS: Instalamos xpm212 con ip address 172.16.5.212

```
#!/bin/bash

# Primero obtenemos un ticket
kinit admin

# Comprobar si se pasó el nombre del host como parámetro
if [ -z "$1" ]; then
  echo "Uso: $0 <nombre_del_host>"
  echo "Ejemplo: $0 pm154"
  exit 1
fi
HOSTNAME=$1
if [ -z "$2" ]; then
  echo "Uso: $0 <nombre_del_host> <ip del host>"
  echo "Ejemplo: $0 $1 172.16.5.212"
  exit 1
fi
IPHOSTNAME=$2
ipa dnsrecord-add iespoveda.com ${HOSTNAME} --a-ip-address=${IPHOSTNAME}
ipa host-add ${HOSTNAME}.iespoveda.com
#ipa-getkeytab -p host/${HOSTNAME}.iespoveda.com -k /etc/krb5.keytab
ipa-getkeytab -p host/${HOSTNAME}.iespoveda.com -k /tmp/${HOSTNAME}.keytab
```

Script ejecutar en MV

Ejecutarlo pasándole como parámetro el nombre del host cliente, por ejemplo xpm212

`setup_clone.sh`

```
#!/bin/bash

# Comprobar si se pasó el nombre del host como parámetro
if [ -z "$1" ]; then
  echo "Uso: $0 <nombre_del_host>"
  echo "Ejemplo: $0 pm154"
  exit 1
fi

HOSTNAME=$1
DOMAIN="iespoveda.com"
REALM="IESPOVEDA.COM"
IPA_SERVER="ipa.iespoveda.com"

echo "[+] Estableciendo el hostname completo..."
hostnamectl set-hostname ${HOSTNAME}.${DOMAIN}

echo "[+] Actualizando paquetes..."
apt update -y

echo "[+] Eliminando paquetes conflictivos con SSSD..."
apt remove --purge -y libnss-ldap libpam-ldap nslcd || true

echo "[+] Instalando paquetes necesarios..."
apt install -y freeipa-client sssd libnss-sss libpam-sss

echo "[+] Creando directorio /etc/ipa..."
mkdir -p /etc/ipa
chown root:root /etc/ipa
chmod 755 /etc/ipa

echo "[+] Descargando certificado de la autoridad (CA)..."
wget -O /etc/ipa/ca.crt http://$IPA_SERVER/ipa/config/ca.crt

echo "[+] Configurando krb5.conf..."
cat > /etc/krb5.conf <<EOF
[libdefaults]
    default_realm = $REALM
    dns_lookup_realm = false
    dns_lookup_kdc = true
    forwardable = true

[realms]
    $REALM = {
        kdc = $IPA_SERVER
        admin_server = $IPA_SERVER
    }

[domain_realm]
    .$DOMAIN = $REALM
    $DOMAIN = $REALM
EOF

echo "[+] Configurando /etc/ipa/default.conf..."
cat > /etc/ipa/default.conf <<EOF
[global]
  domain = $DOMAIN
  hostname = $HOSTNAME.$DOMAIN
  realm = $REALM
```

```
  ipaclient = True
  server = $IPA_SERVER
  ca_cert = /etc/ipa/ca.crt
EOF

echo "[+] Ejecutando ipa-client-install..."
read -s -p "Introduce la contraseña de admin de IPA: " IPA_PASS
echo
echo "$IPA_PASS" | ipa-client-install --hostname=${HOSTNAME}.${DOMAIN} \
  --domain=${DOMAIN} \
  --realm=${REALM} \
  --server=${IPA_SERVER} \
  --unattended \
  --principal=admin \
  --password

echo "[+] Iniciando el servicio krb5-kdc..."
systemctl start krb5-kdc

echo "[+] Obteniendo certificado SSL del servidor IPA..."
echo | openssl s_client -showcerts -connect $IPA_SERVER:443 2>/dev/null | openssl x509 -outform
PEM > ipa_cert.pem

echo "[+] Instalando certificado como CA confiable..."
cp ipa_cert.pem /usr/local/share/ca-certificates/ipa_cert.crt
update-ca-certificates

echo "[+] Copiando keytab (requiere acceso SSH al servidor IPA)..."
#scp /tmp/${HOSTNAME}.keytab root@$IPA_SERVER:/etc/krb5.keytab
scp root@$IPA_SERVER:/tmp/${HOSTNAME}.keytab /etc/krb5.keytab

echo "[+] Configuramos sss."
mkdir -p /etc/sssd
tee /etc/sssd/sssd.conf > /dev/null <<EOF
[sssd]
domains = iespoveda.com
config_file_version = 2
services = nss, pam

[domain/iespoveda.com]
id_provider = ipa
auth_provider = ipa
access_provider = ipa
ipa_domain = iespoveda.com
ipa_server = _srv_, ipa.iespoveda.com
ldap_tls_cacert = /etc/ipa/ca.crt
EOF

echo "[+] Corrigiendo permisos del archivo sssd.conf si existe..."
if [ -f /etc/sssd/sssd.conf ]; then
  chmod 600 /etc/sssd/sssd.conf
  chown root:root /etc/sssd/sssd.conf
fi

echo "[+] Configurando NSS para usar SSSD..."
sed -i '/^passwd:/ s/$/ sss/' /etc/nsswitch.conf
sed -i '/^group:/ s/$/ sss/' /etc/nsswitch.conf
sed -i '/^shadow:/ s/$/ sss/' /etc/nsswitch.conf

# Vamos a trabajar vía NFS
#echo "[+] Configurando PAM para mkhomedir..."
#grep -q pam_mkhomedir.so /etc/pam.d/common-session || echo "session required pam_mkhomedir.so
skel=/etc/skel umask=0022" >> /etc/pam.d/common-session

echo "[+] Reiniciando servicios..."
systemctl restart sssd

echo "[+] kinit con usuario admin..."
kinit admin && klist
```

```
echo "[✔] Cliente FreeIPA configurado correctamente."
# echo, ejemplo: getent passwd arcadio




# Para os el sistema y ponemos en marcha chronyd
echo "[+] Paramos systemd-timesyncd y usmos chrony"
systemctl stop systemd-timesyncd
systemctl disable systemd-timesyncd
systemctl enable --now chronyd
systemctl stop chronyd

echo "[+] configuramos chrony"
cat > /etc/chrony/chrony.conf <<EOF
# Welcome to the chrony configuration file. See chrony.conf(5) for more
confdir /etc/chrony/conf.d
sourcedir /run/chrony-dhcp
sourcedir /etc/chrony/sources.d
keyfile /etc/chrony/chrony.keys
driftfile /var/lib/chrony/chrony.drift
ntsdumpdir /var/lib/chrony
logdir /var/log/chrony
maxupdateskew 100.0
rtcsync
makestep 1 3
leapsectz right/UTC
#servidores
server ipa.iespoveda.com iburst
#
allow 172.16.0.0/16
port 123
#
driftfile /var/lib/chrony/chrony.drift
makestep 1.0 3
EOF

echo "[+] servicio chrony"
systemctl start chronyd
echo " sincronizamos el sistema de manera inmediata"

# -a hace que chronyc se conecte al demonio como administrador (requerido para ciertos
comandos).
# makestep le dice a chronyd que ajuste inmediatamente el reloj del sistema, en vez de ir
corrigiendo gradualmente como suele hacer.
chronyc -a makestep

echo "Vemos que funciona el sistema de hora"
chronyc tracking
chronyc sources
# Cara cambiar la fecha hora:
#date
#date -s "2025-04-24 13:19:00"
#systemctl restart chronyd
systemctl status chrony

echo "[+] Reiniciamos sssd"
systemctl stop sssd
systemctl start sssd


echo "[+] instalamos nfs..."
apt install nfs-common -y

echo "[+] Añadimos la entrada en nfs para que se carguen los homes de los usuarios"
echo -e "\n#\ntruenas:/mnt/truenasdisk/nfsdata /mnt/home nfs defaults 0 0" | tee -a /etc/fstab
> /dev/null

echo "----------------------"
echo "[+] Añdimos qemu-guest-agent para que podamos obtener, entro otras, las direcciones ip de
```

```
las MV que estan activas"
sudo apt install qemu-guest-agent -y
```

Notas

Verificar ticket kerberos

```
klist -f
kdestroy
kinit -f admin
klist -f
```

Verificar ntp server

-- Para verificar los ntp servers podemos hacer:

```
ss -lunp | grep 123"
dig pool.ntp.org +short
traceroute pool.ntp.org
ping -c 3 pool.ntp.org
```

```
nc -zv -u hora.cica.es 123
nc -zv -u pool.netp.org 123
nc -zv -u pool.ntp.org 123
nc -zv -u hora.rediris.es 123
```

```
apt install ntpdate -y
ntpdate -d pool.ntp.org
```

Sincronización automática de la hora mediante NTP

Esta orden activa la sincronización automática de la hora mediante NTP (Network Time Protocol) usando el servicio que ofrece systemd-timesyncd.

```
timedatectl set-ntp true
```

Muestra el estado actual de la fecha y hora del sistema, junto con si NTP está habilitado y si la sincronización está activa.

```
timedatectl status
```

Ejemplo de salida:

```
Local time: Thu 2025-04-25 10:30:00 UTC
Universal time: Thu 2025-04-25 10:30:00 UTC
RTC time: Thu 2025-04-25 10:30:00
Time zone: Etc/UTC (UTC, +0000)
System clock synchronized: yes
NTP service: active
RTC in local TZ: no
```

Comando muestra información detallada sobre el estado de sincronización de tiempo, incluyendo a qué servidor NTP se está conectando, cuándo fue la última sincronización, etc.

```
timedatectl show-timesync --all
```

Desactiva la sincronización automática de la hora

```
timedatectl set-ntp false
```

Este comando sincroniza la hora de tu máquina inmediatamente con un servidor de la red de tiempo pública pool.ntp.org.

```
sudo apt install ntpsec-ntpdate
ntpdate pool.ntp.org
```

a la sincronización automática.

Operaciones en proxmox

Proxmox. Algunos comandos

- Parar la MV en el nodo origen:

```
qm stop <ID_VM>
```

- Copiar el disco: Dependiendo del tipo de almacenamiento, por ejemplo si es una imagen .qcow2 o .raw en /var/lib/vz/images/:

```
rsync -avz /var/lib/vz/images/<ID_VM>/ root@<IP_DESTINO>:/var/lib/vz/images/<ID_VM>/
```

- Copiar la configuración de la VM:

```
scp /etc/pve/qemu-server/<ID_VM>.conf root@<IP_DESTINO>:/etc/pve/qemu-server/
```

- Iniciar la MV en el nodo destino:

```
qm start <ID_VM>
```

- Ver espacio libre general (sistemas de archivos como local)

```
ssh root@<IP_O_NOMBRE_DEL_NODO_DESTINO> df -h
```

- Ver espacio en local-lvm (si usas LVM para almacenamiento)

```
ssh root@<IP_NODO_DESTINO> lvs
```

- O para más detalles: Mostrará el espacio libre en los volúmenes lógicos, como data si usas local-lvm ssh root@<IP_NODO_DESTINO> vgs

Listar máquinas virtuales

```
root@pm153:~# qm list
      VMID NAME                 STATUS     MEM(MB)    BOOTDISK(GB) PID
       211 xpm211               stopped    2048              32.00 0
       213 xpm213               stopped    4096              40.00 0
       217 xpm217               stopped    2048              32.00 0
       218 xpm218               stopped    2048              32.00 0
```

pvecm status

Lo ejecutamos en la consola de un nodo directamente

```
root@pm153:~# pvecm status
Cluster information
-------------------
Name:             ASIR2
Config Version:   10
Transport:        knet
Secure auth:      on
```

- Name: `ASIR2` – Nombre del cluster.
- Config Version: `10` – Versión actual de la configuración del cluster.
- Transport: `knet` – Protocolo de red usado para comunicación entre nodos (en Proxmox moderno, es `knet`).
- Secure auth: `on` – La autenticación segura está activada entre nodos (más seguro).

```
Quorum information
------------------
Date:             Tue May 13 15:05:44 2025
Quorum provider:  corosync_votequorum
Nodes:            6
Node ID:          0x00000001
Ring ID:          1.715
Quorate:          Yes
```

- Quorum provider: `corosync_votequorum` – Servicio encargado de gestionar el quórum.
- Nodes: `6` – Hay 6 nodos actualmente activos en el cluster.
- Node ID: `0x00000001` – ID del nodo donde ejecutaste el comando (el nodo local).
- Ring ID: `1.715` – Identificador de la ronda de comunicación (interno de Corosync).
- Quorate: `Yes` – ¡El cluster tiene quórum! Puede operar con normalidad.

```
Votequorum information
----------------------
Expected votes:   8
Highest expected: 8
Total votes:      6
Quorum:           5
Flags:            Quorate
```

- Expected votes: `8` – El cluster espera un total de 8 votos para estar completo.
- Highest expected: `8` – Máximo de votos esperados históricamente (por configuración).
- Total votes: `6` – Votos reales disponibles ahora (es decir, nodos en línea con voto activo).
- Quorum: `5` – Se necesitan al menos 5 votos para mantener el quórum.
- Flags: `Quorate` – El estado actual del cluster es "quórate", es decir, funcional.

El quórum es esencial para evitar una "división del cerebro" (split-brain), donde varios nodos creen ser el líder. Si no hay quórum, no se pueden hacer cambios en máquinas virtuales, por ejemplo

```
Membership information
----------------------
    Nodeid      Votes Name
0x00000001          1 172.16.5.153 (local)
0x00000002          1 172.16.5.154
0x00000003          1 172.16.5.155
0x00000004          1 172.16.5.156
0x00000006          1 172.16.5.140
0x00000008          1 172.16.5.152
root@pm153:~#
```

- Se listan los nodos actualmente conectados al cluster.
- Cada nodo tiene:
  - Un Node ID hexadecimal único.
  - Votes: Todos tienen 1 voto.
  - Name: Dirección IP del nodo (y se marca cuál es el local).

Importante El total de votos actuales es 6 (cada nodo tiene 1 voto), aunque el cluster espera 8. Esto indica que hay 2 nodos apagados o desconectados, pero como el quórum es 5, y hay 6 votos, todo sigue funcionando.

qm config

Parámetros de configuración de una VM administrada por qm (el gestor de máquinas virtuales de Proxmox que usa QEMU/KVM)

```
root@pm153:~# qm config 211
agent: 1
```

Indica que está activado el QEMU Guest Agent, una herramienta que mejora la integración entre la VM y el host (por ejemplo, para apagarla de forma segura, ver IP, etc.).

```
boot: order=scsi0;ide2;net0
cores: 1
cpu: x86-64-v2-AES
```

La VM tiene 1 núcleo (core) y 1 socket de CPU (es decir, 1 vCPU total).

```
ide2: local:iso/ubuntu-24.10-live-server-amd64.iso,media=cdrom,size=2049278K
memory: 2048
meta: creation-qemu=9.0.2,ctime=1745826303
```

- creation-qemu: versión de QEMU usada para crear la VM.
- ctime: marca de tiempo Unix de creación (puede convertirse para ver la fecha exacta).

```
name: xpm211
net0: virtio=BC:24:11:3D:35:1B,bridge=vmbr0,firewall=1
numa: 0
```

NUMA (Non-Uniform Memory Access) desactivado. Esto suele dejarse en 0 para VMs pequeñas.

```
ostype: l26
scsi0: nfs-vm:211/vm-211-disk-0.qcow2,iothread=1,size=32G
scsihw: virtio-scsi-single
smbios1: uuid=b43f444a-1478-41f1-a04d-c20830473d2f
sockets: 1
vmgenid: ba36889b-451a-4b49-8f86-dc49b3f65409
```

ID único generado para esta VM. Ayuda en operaciones como replicación o restauraciones

Proxmox Información

Campos relevantes que puedes extraer de Proxmox

Del API de Proxmox (/cluster/resources?type=vm) puedes extraer:

- vmid → ID de VM
- name → Nombre de la VM (para connection_name)
- status → Si está running
- node → Nodo en el que corre
- maxmem, maxcpu, uptime, etc.
- IP (solo si QEMU Agent está instalado y activo en la VM)

```
pip install requests
```

```
import requests

# Configuración
PROXMOX_HOST = 'https://172.16.5.154:8006'
USERNAME = 'root@pam'
PASSWORD = 'tu_contraseña'
VERIFY_SSL = False  # Cambiar a True en producción si el certificado es válido

def login():
    url = f'{PROXMOX_HOST}/api2/json/access/ticket'
    data = {
        'username': USERNAME,
        'password': PASSWORD
    }

    response = requests.post(url, data=data, verify=VERIFY_SSL)
    response.raise_for_status()
    result = response.json()['data']

    return {
        'ticket': result['ticket'],
        'csrf_token': result['CSRFPreventionToken'],
        'cookies': {'PVEAuthCookie': result['ticket']}
    }

def get_vms(auth):
    url = f'{PROXMOX_HOST}/api2/json/cluster/resources?type=vm'
    response = requests.get(url, cookies=auth['cookies'], verify=VERIFY_SSL)
    response.raise_for_status()
    return response.json()['data']

def get_vm_ips(auth, node, vmid):
    url = f"{PROXMOX_HOST}/api2/json/nodes/{node}/qemu/{vmid}/agent/network-get-interfaces"
    try:
        response = requests.get(url, cookies=auth['cookies'], verify=VERIFY_SSL)
        if response.status_code == 200:
            data = response.json()['data']
            ips = []
            for iface in data:
                for addr in iface.get('ip-addresses', []):
```

```
                    if addr['ip-address-type'] == 'ipv4' and not addr['ip-
address'].startswith('127.'):
                        ips.append(addr['ip-address'])
            return ips
    except:
        pass
    return []

def main():
    auth = login()
    vms = get_vms(auth)

    for vm in vms:
        print(f"VMID: {vm.get('vmid')}")
        print(f"Name: {vm.get('name')}")
        print(f"Status: {vm.get('status')}")
        print(f"Node: {vm.get('node')}")
        print(f"Max CPU: {vm.get('maxcpu')}")
        print(f"Max MEM: {vm.get('maxmem') / 1024**3:.2f} GB")
        print(f"Uptime: {vm.get('uptime')} s")

        if vm.get('status') == 'running':
            ips = get_vm_ips(auth, vm['node'], vm['vmid'])
            print(f"IP(s): {', '.join(ips) if ips else 'No disponible'}")
        else:
            print("IP: VM no está encendida")

        print("-" * 40)

if __name__ == '__main__':
    main()
```

Ahora lo ejecutamos:

```
guacuser@guacamole:~$ python3 datos_nodo.py
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:1020: InsecureRequestWarning:
Unverified HTTPS request is being made to host '172.16.5.154'. Adding certificate
verification is strongly advised. See: https://urllib3.readthedocs.io/en/latest/advanced-
usage.html#ssl-warnings
  warnings.warn(
/usr/lib/python3/dist-packages/urllib3/connectionpool.py:1020: InsecureRequestWarning:
Unverified HTTPS request is being made to host '172.16.5.154'. Adding certificate
verification is strongly advised. See: https://urllib3.readthedocs.io/en/latest/advanced-
usage.html#ssl-warnings
  warnings.warn(
VMID: 211
Name: xpm211
Status: stopped
Node: pm153
Max CPU: 1
Max MEM: 2.00 GB
Uptime: 0 s
IP: VM no está encendida
----------------------------------------
VMID: 212
Name: xpm212
Status: stopped
Node: pm154
Max CPU: 1
Max MEM: 2.00 GB
Uptime: 0 s
IP: VM no está encendida
----------------------------------------
```

```
VMID: 213
Name: xpm213
Status: stopped
Node: pm153
Max CPU: 1
Max MEM: 4.00 GB
Uptime: 0 s
IP: VM no está encendida
----------------------------------------
VMID: 217
Name: xpm217
Status: stopped
Node: pm153
Max CPU: 1
Max MEM: 2.00 GB
Uptime: 0 s
IP: VM no está encendida
----------------------------------------
VMID: 218
Name: xpm218
Status: stopped
Node: pm153
Max CPU: 1
Max MEM: 2.00 GB
Uptime: 0 s
IP: VM no está encendida
----------------------------------------
VMID: 222
Name: xpm222
Status: stopped
Node: pm156
Max CPU: 1
Max MEM: 2.00 GB
Uptime: 0 s
IP: VM no está encendida
----------------------------------------
```

LDAP. Información

De manera manual:

```
guacuser@guacamole:~$ ldapsearch -x -H ldap://ipa.iespoveda.com -D
"uid=admin,cn=users,cn=accounts,dc=iespoveda,dc=com" -W -b
"cn=groups,cn=accounts,dc=iespoveda,dc=com" "(cn=profesores)"
Enter LDAP Password:
# extended LDIF
#
# LDAPv3
# base <cn=groups,cn=accounts,dc=iespoveda,dc=com> with scope subtree
# filter: (cn=profesores)
# requesting: ALL
#

# profesores, groups, accounts, iespoveda.com
dn: cn=profesores,cn=groups,cn=accounts,dc=iespoveda,dc=com
cn: profesores
description: Grupo de profesores
objectClass: top
objectClass: groupofnames
objectClass: nestedgroup
objectClass: ipausergroup
objectClass: ipaobject
objectClass: posixgroup
objectClass: ipantgroupattrs
ipaUniqueID: 352cea50-298e-11f0-9d92-d8bbc1dcea46
gidNumber: 514000005
ipaNTSecurityIdentifier: S-1-5-21-483169404-2447074518-752674619-1005
member: uid=profesor1,cn=users,cn=accounts,dc=iespoveda,dc=com
member: uid=profesor2,cn=users,cn=accounts,dc=iespoveda,dc=com

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1

</cn=groups,cn=accounts,dc=iespoveda,dc=com>
```

Por medio de programación:

```
sudo apt install python3-pip
pip install ldap3
```

Para extraer datos de usuarios y grupos desde FreeIPA automáticamente, puedes usar ldap3 en Python.

```
from ldap3 import Server, Connection, ALL

LDAP_SERVER = 'ipa.iespoveda.com'
LDAP_USER = 'uid=admin,cn=users,cn=accounts,dc=iespoveda,dc=com'
LDAP_PASS = 'your_password'

def get_group_members(group_name='profesores'):
    server = Server(LDAP_SERVER, get_info=ALL)
    conn = Connection(server, LDAP_USER, LDAP_PASS, auto_bind=True)

    base_dn = 'cn=groups,cn=accounts,dc=iespoveda,dc=com'
    conn.search(base_dn, f'(cn={group_name})', attributes=['member'])

    members_dns = conn.entries[0]['member']
    usernames = []

    for dn in members_dns:
        conn.search(dn, '(objectClass=person)', attributes=['uid'])
        if conn.entries:
            usernames.append(conn.entries[0]['uid'].value)

    conn.unbind()
    return usernames
if __name__ == '__main__':
    miembros = get_group_members('profesores')
    print("Miembros del grupo 'profesores':", miembros)
```

Para ejecutarlo:

```
python ldap_test.py
:~$ python3 ldap_test.py
Miembros del grupo 'profesores': ['profesor1', 'profesor2']
```

Proxmox. IP

Validar si Proxmox obtiene la IP

```
qm guest cmd <vmid> network-get-interfaces</vmid>
[
  {
    "name": "eth0",
    "hardware-address": "xx:xx:xx:xx:xx:xx",
    "ip-addresses": [
      {
        "ip-address": "192.168.1.100",
        "prefix": 24,
        "ip-address-type": "ipv4"
      }
    ]
  }
]
```

Ejemplos:

```
root@pm153:~# qm guest cmd 213 network-get-interfaces
VM 213 is not running
root@pm153:~# qm guest cmd 218 network-get-interfaces
QEMU guest agent is not running
```

En la máquina virtual ejecutamos: `sudo apt install qemu-guest-agent -y` y reboot

```
root@pm153:~# qm guest cmd 218 network-get-interfaces
[
   {
      "hardware-address" : "00:00:00:00:00:00",
      "ip-addresses" : [
         {
            "ip-address" : "127.0.0.1",
            "ip-address-type" : "ipv4",
            "prefix" : 8
         },
         {
            "ip-address" : "::1",
            "ip-address-type" : "ipv6",
            "prefix" : 128
         }
      ],
      "name" : "lo",
      "statistics" : {
         "rx-bytes" : 12034,
         "rx-dropped" : 0,
         "rx-errs" : 0,
         "rx-packets" : 120,
         "tx-bytes" : 12034,
         "tx-dropped" : 0,
         "tx-errs" : 0,
         "tx-packets" : 120
      }
   },
   {
      "hardware-address" : "bc:24:11:4b:b3:aa",
      "ip-addresses" : [
         {
            "ip-address" : "172.16.5.218",
            "ip-address-type" : "ipv4",
            "prefix" : 16
         },
         {
            "ip-address" : "fe80::be24:11ff:fe4b:b3aa",
            "ip-address-type" : "ipv6",
            "prefix" : 64
         }
      ],
      "name" : "ens18",
      "statistics" : {
         "rx-bytes" : 54100,
         "rx-dropped" : 0,
         "rx-errs" : 0,
         "rx-packets" : 580,
         "tx-bytes" : 26396,
         "tx-dropped" : 0,
         "tx-errs" : 0,
         "tx-packets" : 177
      }
   }
]
```

Planificación clases. Pasos a realizar por el profesor

Algunas operaciones de inteŕes

para ser template

```
qm template <vmid>
</vmid>
```

Dejar de ser template

```
root@pm154:~# qm set 212 --template 0
update VM 212: -template 0
```

Usando TEMPLATES para generar máquinas virtuales

Ejecutamos en el nodo en el que vamos a tener el template. Tener en cuenta:

- El identificativo que le vamos a dar, el id, VMID
- Variables de etrada
- Será almacenado en nfs-vm. En truenas vía NFS

`crear_template_ubuntu_cloudinit.sh`

```
#!/bin/bash

# CONFIGURACIÓN
VMID=9000
VMNAME="ubuntu-cloudinit-template"
MEM=2048
CORES=2
BRIDGE="vmbr0"
STORAGE="nfs-vm"
DISK_SIZE="8G"
NETIP="172.16.5.245/16"
NETGW="172.16.0.1"
NETDNS="172.16.5.110"
UBUNTU_VERSION="jammy"
IMG_URL="https://cloud-images.ubuntu.com/${UBUNTU_VERSION}/current/${UBUNTU_VERSION}-server-
cloudimg-amd64.img"
IMG_NAME="${UBUNTU_VERSION}-server-cloudimg-amd64.img"
TEMPLATE_PATH="/var/lib/vz/template"

# 1. Descargar imagen cloud-init oficial
echo "[+] Descargando imagen Ubuntu Cloud ($UBUNTU_VERSION)..."
mkdir -p $TEMPLATE_PATH && cd $TEMPLATE_PATH
wget -N $IMG_URL

# 2. Crear VM base
echo "[+] Creando VM $VMID ($VMNAME)..."
qm create $VMID --name $VMNAME --memory $MEM --cores $CORES --net0 virtio,bridge=$BRIDGE

# 3. Importar disco
echo "[+] Importando disco en $STORAGE..."
qm importdisk $VMID $IMG_NAME $STORAGE

# 4. Conectar el disco importado como SCSI
echo "[+] Conectando disco importado como SCSI..."
qm set $VMID --scsihw virtio-scsi-pci --scsi0 ${STORAGE}:vm-${VMID}-disk-0

# 5. Añadir disco Cloud-Init
echo "[+] Añadiendo disco Cloud-Init..."
qm set $VMID --ide2 ${STORAGE}:cloudinit

# 6. Configuración de arranque y consola
echo "[+] Configurando disco de arranque y consola..."
qm set $VMID --boot c --bootdisk scsi0 --serial0 socket --vga serial0

# 7. Activar QEMU Guest Agent
echo "[+] Activando QEMU Guest Agent..."
qm set $VMID --agent enabled=1

# 8. Configurar valores por defecto cloud-init (opcional)
qm set $VMID --ciuser ubuntu --cipassword ubuntu
```

```
# IP fija + gateway
qm set $VMID --ipconfig0 "ip=$NETIP,gw=$NETGW"

# Servidor DNS
qm set $VMID --nameserver $NETDNS

# deshabilito pxe
qm set $VMID --scsihw virtio-scsi-pci --scsi0 nfs-vm:$VMID/vm-$VMID-disk-0.raw
qm set $VMID --boot order=scsi0 --bootdisk scsi0

#qm set $VMID --boot c
#qm set $VMID --bootdisk scsi0


# 8. Iniciamos la máquina virtual
qm start $VMID

# 9. Abre la consola desde la GUI o CLI:
echo "Inicia sesión como ubuntu o cloud-user"
qm terminal $VMID
```

Puedo ver los datos en, suponemos que he generado el id 245:

```
root@pm154:~# qm config 245
agent: enabled=1
boot: order=scsi0
cipassword: **********
ciuser: ubuntu
cores: 2
ide2: nfs-vm:245/vm-245-cloudinit.qcow2,media=cdrom
ipconfig0: ip=172.16.5.245/16,gw=172.16.0.1
memory: 2048
meta: creation-qemu=9.0.2,ctime=1747998403
name: ubuntu-cloudinit-template
nameserver: 172.16.5.110
net0: virtio=BC:24:11:1F:AA:60,bridge=vmbr0
scsi0: nfs-vm:245/vm-245-disk-0.raw,size=2252M
scsihw: virtio-scsi-pci
serial0: socket
smbios1: uuid=50ec7007-ae35-4faa-991a-6e3c299721c2
vga: serial0
vmgenid: c83c255a-560c-45af-9cde-240b26991086
```

Ahora dentro de la máquina virtual, 245 en nuestro caso, ejecutamos:

```
# Actualiza sistema
sudo apt update && sudo apt install cloud-init qemu-guest-agent -y

# Habilita qemu-guest-agent
sudo systemctl enable --now qemu-guest-agent
```

Añadimos nuestro script, el que tiene que ejecutarse tras iniciarse el clone

```
# Añadir tu script FreeIPA, es el que hemos usado para ejecutar en las máquinas virtuales una
vez que se generan.
sudo vi /usr/local/bin/setup_clone.sh
sudo chmod +x /usr/local/bin/setup_clone.sh
```

Ahora volvempos al proceso de generar la plantilla:

```
sudo cloud-init clean --logs
sudo shutdown now
```

Volvemos a nuestro nodo

```
qm template $VMID
```

Puede ocurrir que, a parte de crear el template, nos encontremos:

```
/usr/bin/chattr: Operation not supported while reading flags on /mnt/pve/nfs-
vm/images/245/base-245-disk-0.raw
command '/usr/bin/chattr +i /mnt/pve/nfs-vm/images/245/base-245-disk-0.raw' failed: exit code
1
ocurre porque el comando chattr +i (que pone el archivo como inmutable) no es soportado en
sistemas de ficheros NFS (Network File System). chattr funciona en sistemas de ficheros
locales como ext4, xfs, etc., pero NFS no admite esas operaciones.
```

Ahora vamos a crear nuestro primer clone.:

```
#!/bin/bash

# Configuraciones base
STORAGE="nfs-vm"
BRIDGE="vmbr0"

read -p "Introduce el id del template base (ej: 900): " TEMPLATE_ID
read -p "Introduce el hostname de la nueva VM (ej: pm154): " HOSTNAME
read -p "Introduce el VMID de la nueva VM (ej: 9010): " NEW_VMID

read -s -p "Introduce la contraseña del usuario root (no se mostrará): " PASSWORD
echo

# Pregunta si usar DHCP o IP estática
read -p "¿Quieres usar DHCP? (s/n): " DHCP_CHOICE

if [[ "$DHCP_CHOICE" =~ ^[Ss]$ ]]; then
  IPCONFIG="ip=dhcp"
else
  read -p "Introduce la IP estática (ej: 192.168.1.100): " STATIC_IP
  read -p "Introduce la máscara CIDR (ej: 24 para /24): " MASK
  read -p "Introduce el gateway (ej: 192.168.1.1): " GATEWAY
  IPCONFIG="ip=${STATIC_IP}/${MASK},gw=${GATEWAY}"
fi

echo "[+] Clonando VM desde plantilla $TEMPLATE_ID..."
qm clone $TEMPLATE_ID $NEW_VMID --name $HOSTNAME --full

echo "[+] Configurando red: $IPCONFIG"
qm set $NEW_VMID --ipconfig0 "$IPCONFIG"

echo "[+] Configurando usuario root con cloud-init..."
qm set $NEW_VMID --ciuser root --cipassword "$PASSWORD"

echo "[+] Asignando hostname..."
#qm set $NEW_VMID --hostname "$HOSTNAME"

echo "[+] Activando QEMU Guest Agent..."
qm set $NEW_VMID --agent enabled=1

echo "[+] Iniciando VM $NEW_VMID..."
echo "Cuando se inicie TIENES QUE EJECUTAR EL SCRIPT, QUE YA ESTÁ EN EL CLONE,
/usr/local/bin/setup_clone.sh"
qm start $NEW_VMID

echo "[✔] VM clonada y configurada: ID $NEW_VMID - Hostname: $HOSTNAME"
```

IMPORTANTE: Antes de ejecutar `/usr/local/bin/setup_clone.sh` en el clone generado tenemos que ejecutar en el **ipa server** por ejemplo: `./cliente_ipa.sh xpm246 172.16.5.246`. Esto es:

```
kinit admin
```

```
[root@ipa ~]# cat cliente_ipa.sh
#!/bin/bash

# Comprobar si se pasó el nombre del host como parámetro
if [ -z "$1" ]; then
  echo "Uso: $0 <nombre_del_host>"
  echo "Ejemplo: $0 pm154"
  exit 1
fi
HOSTNAME=$1
if [ -z "$2" ]; then
  echo "Uso: $0 <nombre_del_host> <ip del="" host="">"
  echo "Ejemplo: $0 $1 172.16.5.212"
  exit 1
fi
IPHOSTNAME=$2
ipa dnsrecord-add iespoveda.com ${HOSTNAME} --a-ip-address=${IPHOSTNAME}
ipa host-add ${HOSTNAME}.iespoveda.com
#ipa-getkeytab -p host/${HOSTNAME}.iespoveda.com -k /etc/krb5.keytab
ipa-getkeytab -p host/${HOSTNAME}.iespoveda.com -k /tmp/${HOSTNAME}.keytab

</ip></nombre_del_host></nombre_del_host>
```

MV Desde nfs a local

Vamos a tratar de poner en local una máquina virtual a la que se puede acceder desde nfs para que pueda ser ejecutada en el nodo de manera local
Será más rápido
Damos permiso de ejecución a un usuario en particular
Seguimos con IPA server y el usuario montará su home desde nfs

Habilitar images en el almacenamiento local

Edita el archivo /etc/pve/storage.cfg y busca esta sección:

```
dir: local
    path /var/lib/vz
    content iso,vztmpl,backup
```

Cámbiala a:

```
dir: local
    path /var/lib/vz
    content iso,vztmpl,backup,images
```

Luego guarda y reinicia el servicio de Proxmox (o solo intenta de nuevo) Ejemplo, tenemos:

```
root@pm153:~# cat /etc/pve/storage.cfg
dir: local
 path /var/lib/vz
 content backup,vztmpl,iso

lvmthin: local-lvm
 thinpool data
 vgname pve
 content images,rootdir

nfs: nfs-vm
 export /mnt/truenasdisk/nfsvm
 path /mnt/pve/nfs-vm
 server 172.16.5.111
 content images
 nodes pm152,pm153,pm155,pm156,pm154
 prune-backups keep-all=1
```

así:

```
dir: local
 path /var/lib/vz
 content backup,vztmpl,iso,images
```

Nota: Si tuvimos un error de una mv que hemos puesto en el directorio antes de añadir "images" haríamos por ejemplo
"qm set 153212 --scsi0 local:153212/vm-212-disk-0.qcow2"
y obtndríamos: "update VM 153212: -scsi0 local:153212/vm-212-disk-0.qcow2"

Script para cambiar de nfs a local y dar permiso de ejecución

```
#!/bin/bash

# Uso: ./migrar_vm_desde_nodo.sh <ID_ORIGEN> <NODO_ORIGEN> <ID_NUEVO> <USUARIO>
# Ejemplo: ./migrar_vm_desde_nodo.sh.sh 212 pm154 1212 arcadio

set -e

VMID_ORIG="$1"
NODO_ORIGEN="$2"
VMID_NUEVO="$3"
USUARIO="$4"
DOMAIN="iespoveda.com"

# Validación de parámetros
if [[ -z "$VMID_ORIG" || -z "$NODO_ORIGEN" || -z "$VMID_NUEVO" || -z "$USUARIO" ]]; then
  echo "Uso: $0 <ID_ORIGEN> <NODO_ORIGEN> <ID_NUEVO> <USUARIO>"
  exit 1
fi

ORIG_CONF="/etc/pve/nodes/$NODO_ORIGEN/qemu-server/$VMID_ORIG.conf"
NEW_CONF="/etc/pve/qemu-server/$VMID_NUEVO.conf"

if [[ ! -f "$ORIG_CONF" ]]; then
  echo "[ERROR] No se encontró la configuración de la VM $VMID_ORIG en el nodo $NODO_ORIGEN."
  exit 1
fi

echo "[INFO] Clonando VM $VMID_ORIG de $NODO_ORIGEN como nueva VM $VMID_NUEVO en este nodo..."

# Extraer configuración básica
NAME=$(grep -Po '^name: \K.*' "$ORIG_CONF" || echo "vm$VMID_NUEVO")
MEMORY=$(grep -Po '^memory: \K.*' "$ORIG_CONF" || echo "2048")
CORES=$(grep -Po '^cores: \K.*' "$ORIG_CONF" || echo "2")
SOCKETS=$(grep -Po '^sockets: \K.*' "$ORIG_CONF" || echo "1")
NET0=$(grep -Po '^net0: \K.*' "$ORIG_CONF" || echo "virtio,bridge=vmbr0")

# Crear nueva VM
qm create "$VMID_NUEVO" --name "$NAME" --memory "$MEMORY" --cores "$CORES" --sockets "$SOCKETS"
--net0 "$NET0"

# Detectar disco original
DISK_LINE=$(grep -Po '^(scsi|ide|sata|virtio)0: \K[^,]+' "$ORIG_CONF")
if [[ -z "$DISK_LINE" ]]; then
  echo "[ERROR] No se encontró disco principal en la configuración original."
  exit 1
fi

STORAGE_NAME=$(echo "$DISK_LINE" | cut -d: -f1)
DISK_RELATIVE_PATH=$(echo "$DISK_LINE" | cut -d: -f2)
DISK_NAME=$(basename "$DISK_RELATIVE_PATH")

SRC_DISK="/mnt/pve/$STORAGE_NAME/images/$DISK_RELATIVE_PATH"
DST_DIR="/var/lib/vz/images/$VMID_NUEVO"
DST_DISK="$DST_DIR/$DISK_NAME"

echo "[INFO] Copiando disco desde: $SRC_DISK"
mkdir -p "$DST_DIR"
```

```
# Verificar espacio libre
REQUIRED_SIZE=$(qemu-img info --output=json "$SRC_DISK" | grep 'virtual-size' | sed 's/[^0-9]*\
([0-9]\+\).*/\1/')

AVAILABLE=$(df --output=avail -B1 /var/lib/vz | tail -n1)

if [[ "$REQUIRED_SIZE" -ge "$AVAILABLE" ]]; then
  echo "[ERROR] No hay suficiente espacio disponible en /var/lib/vz para copiar el disco."
  exit 1
fi

cp "$SRC_DISK" "$DST_DISK"

# Conectar disco a la VM nueva
qm set "$VMID_NUEVO" --scsi0 "local:$VMID_NUEVO/$DISK_NAME"

# Asignar permisos al usuario
pveum user add "$USUARIO@pve" 2>/dev/null || true
pveum aclmod /vms/$VMID_NUEVO --user "$USUARIO@pve" --role PVEVMUser

echo "[✅ OK] La VM $VMID_NUEVO fue creada localmente y clonada desde $VMID_ORIG. Disco copiado
y permisos asignados a $USUARIO."


# Registrar la nueva VM en el IPA Server
#echo "[INFO] Registrando la nueva VM $VMID_NUEVO en el IPA Server..."
#ipa host-add "$VMID_NUEVO.$IPA_SERVER" --ip-address="<IP_DE_LA_VM>" || {
#  echo "[ERROR] No se pudo registrar la VM en el IPA Server."
#  exit 1
#}
# Registrar la nueva VM en el IPA Server (sin IP)
echo "[INFO] Registrando la nueva VM $VMID_NUEVO en el IPA Server..."
ipa host-add "$VMID_NUEVO.$DOMAIN" || {
  echo "[ERROR] No se pudo registrar la VM en el IPA Server."
  exit 1
}


echo "[✅ OK] La VM $VMID_NUEVO ha sido registrada en IPA Server."

# Actualiza la VM con el nuevo disco
qm set "$VMID_NUEVO" --scsi0 "local:$VMID_NUEVO/$DISK_NAME"

# Configurar la VM para que arranque correctamente
qm set "$VMID_NUEVO" --scsihw virtio-scsi-pci
qm set "$VMID_NUEVO" --bootdisk scsi0
qm set "$VMID_NUEVO" --boot order=scsi0
```

Template desde la MV creada anteriormente

También **podemos crear templates desde nuestras máquinas virtuales para trabajar con clones posteriormente.**

Consideraciones:

- machine-id
- Direcciones ip
- etc.

Problemas

Estamos registrando en ipa server la máquina virtual, lo cual está bien para trabajar con ella pero **no si la vamos a hacer template** por lo que tenemos que separar nuestro proceso.

En servidor IPA antes de generar los clones

Verificaciones previas

Verificamos que podemos hacer:

```
kinit admin
```

El servidor IPA debe tener su CA funcionando correctamente

```
ipa cert-show 0x1
[root@ipa ~]# ipa cert-show
Serial number:
ipa: ERROR: Could not get Serial number interactively
[root@ipa ~]# ipa cert-show 0x1
  Issuing CA: ipa
  Certificate: MIIETjCCAragA.....................0pFOIcIgXbE=
  Subject: CN=Certificate Authority,O=IESPOVEDA.COM
  Issuer: CN=Certificate Authority,O=IESPOVEDA.COM
  Not Before: Wed Mar 26 13:17:25 2025 UTC
  Not After: Sun Mar 26 12:17:25 2045 UTC
  Serial number: 1
  Serial number (hex): 0x1
  Revoked: False
```

y/o

```
ipa-getcert list
```

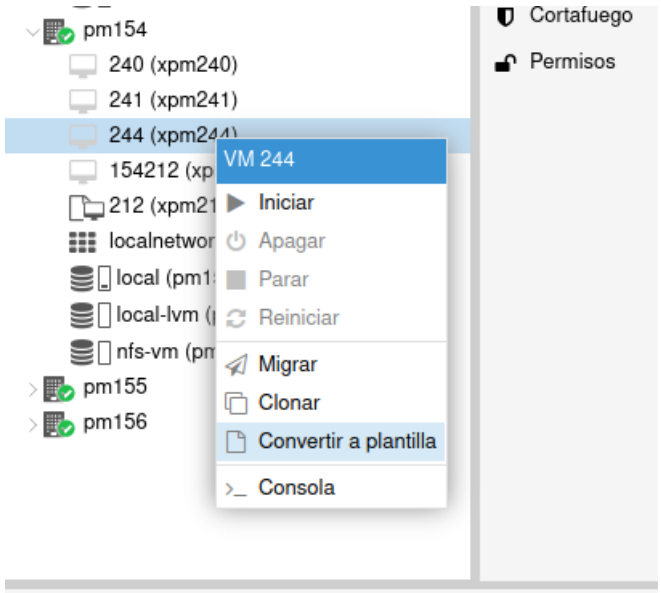
El servicio pki-tomcatd esté activo:

```
systemctl status pki-tomcatd@pki-tomcat
```

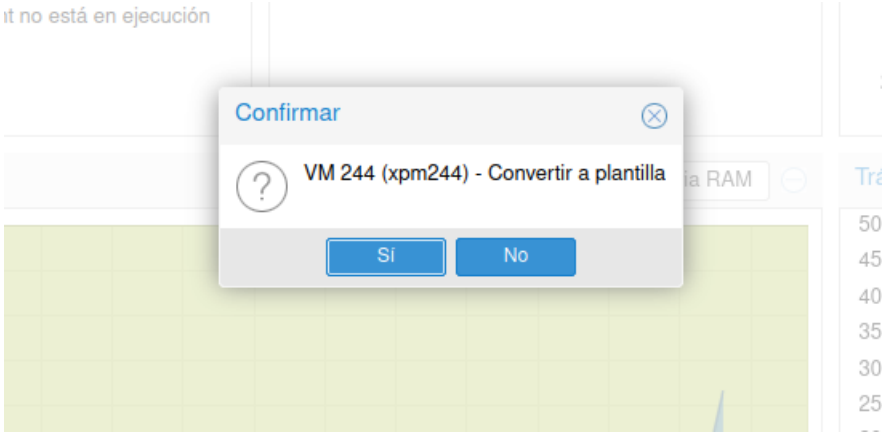
Verifica que los servicios de IPA estén activos:

```
[root@ipa ~]# ipactl status
Directory Service: RUNNING
krb5kdc Service: RUNNING
kadmin Service: RUNNING
named Service: RUNNING
httpd Service: RUNNING
ipa-custodia Service: RUNNING
pki-tomcatd Service: RUNNING
ipa-otpd Service: RUNNING
ipa-dnskeysyncd Service: RUNNING
ipa: INFO: The ipactl command was successful
```

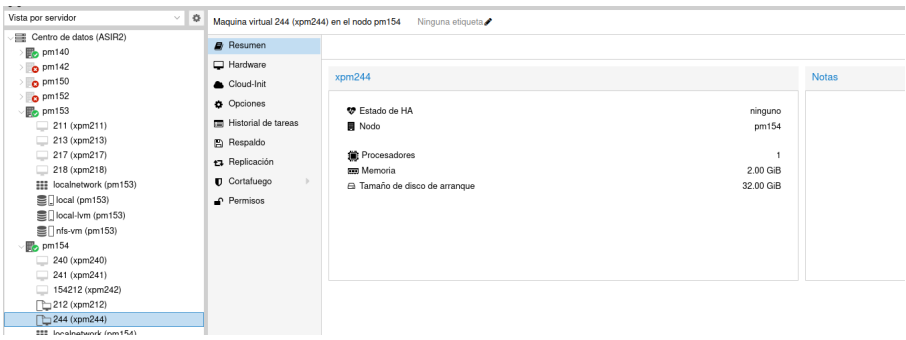
Ahora convertimos la máquina virtual en plantilla:



Seleccionamos nuestra MV para convertirla en plantilla



Proceso de conversión



Plantilla generada

Clonamos la plantilla

Ejemplo:

```
qm clone 244 154244 --name xpm154244 --full
root@pm154:~# qm clone 244 154244 --name xpm154244 --full
create full clone of drive scsi0 (nfs-vm:244/base-244-disk-0.qcow2)
Formatting '/mnt/pve/nfs-vm/images/154244/vm-154244-disk-0.qcow2', fmt=qcow2
cluster_size=65536 extended_l2=off preallocation=metadata compression_type=zlib
size=34359738368 lazy_refcounts=off refcount_bits=16
transferred 0.0 B of 32.0 GiB (0.00%)
...
...
transferred 32.0 GiB of 32.0 GiB (100.00%)
transferred 32.0 GiB of 32.0 GiB (100.00%)
```

Luego hacemos:

```
root@pm154:~# qm set 154244 --ipconfig0="ip=172.16.5.244/16,gw=172.16.5.0"
update VM 154244: -ipconfig0 ip=172.16.5.244/16,gw=172.16.5.0
```

Posterior a template

Primero le damos el hostname, ejemplo:

```
hostnamectl set-hostname xpm154244
```

Laboratorio clase

Partimos de nuestro json `vms_a_migrar.json` y nos encontramos en el nodo en el que hemos definido nuestras máquinas virtuales

```
[
  {
    "vmid": 101,
    "nodo_origen": "pm153",
    "nodo_destino": "pm154",
    "nuevo_vmid": 154101,
    "usuario": "vmuser1"
  },
  {
    "vmid": 102,
    "nodo_origen": "pm153",
    "nodo_destino": "pm155",
    "nuevo_vmid": 155102,
    "usuario": "vmuser2"
  }
]
```

Script: migrar_`migrar_varias_vms.sh` **NOTA: ES NECESARIO** TENER EL SCRIPT: *migrar_vm_desde_nodo.sh* en el hme del root de los nodos destinos.

```
#!/bin/bash

JSON_FILE="vms_a_migrar.json"

if ! command -v jq &>/dev/null; then
  echo "[ERROR] jq no está instalado. Instálalo con: apt install jq"
  exit 1
fi

if [[ ! -f "$JSON_FILE" ]]; then
  echo "[ERROR] No se encontró el archivo $JSON_FILE"
  exit 1
fi

jq -c '.[]' "$JSON_FILE" | while read -r entry; do
  VMID=$(echo "$entry" | jq -r '.vmid')
  NODO_ORIGEN=$(echo "$entry" | jq -r '.nodo_origen')
  NODO_DESTINO=$(echo "$entry" | jq -r '.nodo_destino')
  NUEVO_VMID=$(echo "$entry" | jq -r '.nuevo_vmid')
  USUARIO=$(echo "$entry" | jq -r '.usuario')

  echo "[INFO] Migrando VM $VMID desde $NODO_ORIGEN a $NODO_DESTINO como $NUEVO_VMID (usuario:
$USUARIO)..."

  ssh root@$NODO_DESTINO "./migrar_vm_desde_nodo.sh $VMID $NODO_ORIGEN $NUEVO_VMID $USUARIO"


  echo "[INFO] Finalizada la migración de $VMID -> $NUEVO_VMID"
  echo "------------------------------------------------------------"
done
```

Uso mediante cron

Por ejemplo ejecución diaria a las una de la mañana:

```
0 1 * * * /ruta/al/script/migrar_varias_vms.sh >> /var/log/migrar_vms.log 2>&1
```

Otros datos de gestión

Configurar la autenticación en Proxmox

- Abre la interfaz web de Proxmox.
- Ve a Datacenter → Permissions → User.
- En la sección de "User", puedes agregar a un usuario del IPA para otorgar permisos de acceso.
- Para asignar permisos
para ejecutar una VM
  - Selecciona el usuario del IPA
  - Asocia el rol correspondiente (por ejemplo, PVEAdmin, PVEUser, etc.). Esto se puede hacer en la interfaz web de Proxmox para permitir que el usuario ejecute máquinas virtuales.

Dar permisos para ejecutar la MV clonada

Opciones:
1. Asignar permisos al usuario para ejecutar la VM clonada:
En la interfaz web de Proxmox:
- Ve a la pestaña Datacenter → Permissions → Add.
- Selecciona el usuario (en este caso, el usuario autenticado a través del IPA).
- Asigna el rol adecuado (por ejemplo, PVEAdmin o PVEUser, dependiendo de si deseas dar acceso completo o solo permisos limitados).
- En el campo Path, selecciona la máquina virtual (o el nodo o grupo de nodos donde se encuentra la VM clonada).
- Configura los permisos para "VM.Audit", "VM.PowerMgmt", o similar dependiendo de qué acceso quieras que tenga el usuario. Si el usuario debe poder iniciar y detener la VM, asegúrate de que tenga el permiso VM.PowerMgmt.

2. Configurar permisos desde la línea de comandos (opcional):

```
pveum acl modify /vms/<VMID> -user <usuario_ipa> -role <role>
```

```
# Darle rol de "usuario de VM" (puede arrancar, parar, usar consola)
pveum aclmod /vms/153212 --user arcadio@pve --role PVEVMUser
```

Por ejemplo: Le da al usuario usuario_ipa el rol PVEUser sobre la VM con el ID 101

```
pveum acl modify /vms/101 -user usuario_ipa@yourdomain.com -role PVEUser
```

PVEUser otorga permisos para controlar la máquina virtual (iniciar, detener, etc.).
3. Quitarle el permiso (revocar acceso)

```
pveum acldel /vms/153212 --user arcadio@pve
```

Controlar tiempos de acceso

Duración determinada

```
#!/bin/bash
VMID="$1"
USUARIO="$2"
DURACION_MINUTOS="$3"

# Dar permiso
echo "[+] Otorgando acceso a $USUARIO para la VM $VMID por $DURACION_MINUTOS minutos..."
pveum aclmod /vms/$VMID --user ${USUARIO}@pve --role PVEVMUser

# Esperar y luego revocar
sleep $((DURACION_MINUTOS * 60))

echo "[+] Revocando acceso a $USUARIO..."
pveum acldel /vms/$VMID --user ${USUARIO}@pve
```

Ejemplo uso:

```
./acceso_temporal.sh 153212 arcadio 30
```

A través de crons

permitir_acceso.sh

```
#!/bin/bash
VMID=$1
USUARIO=$2

pveum aclmod /vms/$VMID --user ${USUARIO}@pve --role PVEVMUser
echo "[+] Permiso otorgado a $USUARIO para VM $VMID"
```

revocar_acceso.sh

```
#!/bin/bash
VMID=$1
USUARIO=$2

pveum acldel /vms/$VMID --user ${USUARIO}@pve
echo "[+] Permiso revocado a $USUARIO para VM $VMID"
```

Y luego lo programas así:

```
# A las 8:00 habilitamos acceso
0 8 * * * /root/permitir_acceso.sh 153212 arcadio

# A las 20:00 revocamos acceso
0 20 * * * /root/revocar_acceso.sh 153212 arcadio
```

Nota importante

ACLs en Proxmox Cluster: son globales

Cuando asignas permisos con `pveum aclmod`, los permisos se aplican en todo el cluster (porque las ACLs se almacenan en el sistema distribuido **pmxcfs**).

Para el futuro

¿Qué añadir?

- Backup y Recuperación
  - TrueNas. Sería servidor de respaldo
  - IPA sever. Sería servidor de respaldo
  - Proxmox cloud
- Consideraciones de Red Interna
  - Uso dhcp server con Ipa server y su DNS
  - VLANS para las tres redes importantes del sistema
- Monitorización y Troubleshooting Básico: monitorizar el estado del sistema
  - Recursos en nodos Proxmox
  - Espacio en TrueNAS
  - Estado de servicios como IPA, SSSD, NFS